

## THESIS / THÈSE

### MASTER EN SCIENCES INFORMATIQUES

#### Performance de réseau

Dewandeleer, Daniel; Liette, Nadine

*Award date:*  
1986

*Awarding institution:*  
Université de Namur

[Link to publication](#)

#### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

#### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

FACULTES UNIVERSITAIRES  
NOTRE - DAME DE LA PAIX  
NAMUR  
INSTITUT D'INFORMATIQUE

Titre du mémoire :

# PERFORMANCE DE RESEAU.

Daniel DEWANDELEER.  
Nadine LIETTE.

Stage effectué  
à la Générale de Banque.

Directeur Académique :  
**Mme M. NOIRHOMME**, Professeur  
aux FUNDP, Namur.

Responsable de stage :  
**Mr. R. HEYMANS**,  
Générale de Banque.

Mémoire présenté en vue de  
l'obtention du titre de  
**LICENCIE ET MAITRE** en  
informatique.

Année Académique 1985-1986.

*"Savoir que l'on sait ce que l'on sait, et savoir que l'on ne sait pas ce que l'on ne sait pas : voilà la véritable science."*

CONFUCIUS.

Doctrines ; Le Lun-Yu.

*"... diviser chacune des difficultés que j'examinerais en autant de parcelles qu'il se pourrait et qu'il serait requis pour les mieux résoudre."*

DESCARTES, Discours de la  
Méthode, seconde partie, 1637.

Avant toute chose, nous voulons adresser nos vifs remerciements à **Mme. M. NOIRHOMME**, ainsi qu'à **Mr. J. BARRETO** pour l'aide qu'ils nous ont accordée dans la réalisation de ce mémoire.

Nous voulons également remercier **Mr. R. HEYMANS** pour l'accueil qu'il nous a réservé et les conseils qu'il nous a donnés lors de notre stage à la Générale de Banque.

Nous n'oublierons pas non plus **Mr. P. DE KEYSER** ainsi que **Mr. L. HEMELAER** de la Générale de Banque, pour la collaboration active dont ils nous ont gratifiés.

Enfin, nous saluons également toutes les personnes qui ont contribué de près ou de loin à la réalisation de notre mémoire.

## INTRODUCTION.

*"Une introduction doit toujours à la fois ouvrir le lecteur à ce qui va suivre, tout en lui donnant la mesure ... le tempo de ce qui va suivre"*

J. Berleur

Nous avons voulu que ce mémoire soit un reflet *chronologique* de l'ensemble des problèmes auxquels nous avons été confrontés lors de notre stage. De cette manière, nous espérons que le lecteur se trouvera un peu dans notre situation au début du mois de septembre 1985 et qu'il fera, un an plus tard, le même chemin que nous.

Nous savions à l'époque que nous serions amenés à étudier la performance du réseau de la Générale de Banque, réseau qui nous était totalement inconnu. Nous souhaitions non seulement observer son comportement, mais également l'expliquer.

Nous nous sommes rapidement rendu compte que l'étude des performances d'un réseau d'ordinateurs, tel que le réseau bancaire qui nous occupe ici, n'est pas un problème simple étant donné le nombre de paramètres qui influencent son comportement. Les responsables du réseau de la Générale de Banque possèdent une bonne idée intuitive de ses performances, par la prise d'un échantillon restreint de mesures parmi celles qui sont fournies par les utilitaires de gestion livrés par le constructeur. Néanmoins, une étude systématique de ces performances n'a jamais été effectuée jusqu'ici et c'est ce que nous avons voulu réaliser dans le cadre de ce mémoire.



Quelle a été notre démarche ?

Nous avons parcouru différentes étapes auxquelles correspondent en fait les chapitres qui composent ce travail.

Dans un premier temps, nous nous sommes intéressés à la structure générale du réseau ainsi qu'à son fonctionnement global, tout en étudiant les divers outils qui seraient mis à notre disposition, de manière à acquérir une connaissance suffisante de son exploitation.

C'est ainsi que dans le premier chapitre :

- nous rappelons d'abord les principes élémentaires des réseaux à commutation par paquets.
- nous donnons ensuite une description du réseau et nous expliquons son fonctionnement.
- nous décrivons enfin les fichiers de statistiques fournis par une procédure du constructeur du réseau, ainsi que la méthode pour les obtenir.

Dans un deuxième temps, nous avons décidé de classer l'ensemble des variables à mesurer, de même que toute une série de variables explicatives qui, dans notre esprit, étaient de nature à favoriser la compréhension du comportement du réseau. Nous espérons que cette liste soit la plus exhaustive possible. Le lecteur trouvera également en fin de chapitre des tableaux récapitulatifs quant au rapport entre les variables à expliquer et les variables explicatives.

Après avoir étudié le réseau de manière statique, nous avons essayé dans un troisième temps, de saisir son comportement dynamique. A cette fin, nous avons tenté de dégager plusieurs groupes d'utilisateurs que nous jugions responsables de la charge fournie au réseau. Cette recherche nous a amenés à prendre trois décisions importantes pour la suite de notre étude :

- d'abord, nous n'avons retenu que trois grands groupes d'utilisateurs sur les treize initialement dégagés. Nous les avons sélectionnés en fonction de leur importance relative par rapport à la charge totale du réseau.
- ensuite, pour chaque groupe d'utilisateurs retenu, nous sommes volontairement limités à étudier leur comportement aux périodes critiques, c'est-à-dire aux heures les plus chargées de la journée (Signalons que pour chaque groupe, il ne s'agit pas forcément des mêmes plages horaires).
- enfin, nous avons adopté l'hypothèse simplificatrice, qu'à l'intérieur d'une plage horaire de charge maximale pour un groupe d'utilisateurs donné, toute la charge observée lui était totalement imputable.

Ces décisions exposées au chapitre trois, sont destinées à simplifier l'étude des variables à expliquer, détaillée au chapitre suivant.

Dans un quatrième temps, nous avons effectué un ensemble de mesures concernant les variables à expliquer et nous les avons rassemblées dans le chapitre quatre. Pour chaque

variable à expliquer, nous abordons les problèmes rencontrés et nous exposons la méthode adoptée pour la mesurer. En outre, nous rassemblons une série de mesures effectuées, que nous utiliserons lors de leur exploitation statistique.

Dans un cinquième temps, nous avons essayé d'appliquer une méthode statistique (en l'occurrence les modèles linéaires normaux) afin de dégager pour chaque variable à expliquer un ensemble significatif de variables explicatives parmi celles que nous avons rassemblées dans le chapitre deux.

Dans un sixième temps, nous regroupons toutes les conclusions de notre étude.

Le lecteur trouvera en tête de chaque chapitre un résumé succinct de son contenu. En annexe, il consultera la description détaillée des fichiers statistiques utilisés pour mesurer les variables, les programmes, les tableaux et les graphiques sur le comportement général du réseau ainsi que les modèles linéaires normaux pour chacune des variables étudiées.



## CHAPITRE 1

### DESCRIPTION DU RESEAU DE LA GENERALE DE BANQUE.

-----

#### Résumé :

L'objectif de ce chapitre est de décrire le réseau de la Générale de Banque.

Dans un premier temps, nous rappelons quelques notions de base des réseaux à commutation par paquets afin de comprendre le fonctionnement du réseau particulier que nous avons analysé.

Après avoir décrit globalement le réseau, nous présentons en détail la composition des éléments qu'il renferme (Noeud SL-10, hardware et software). Ensuite, nous développons la partie relative au fonctionnement et au contrôle du réseau et nous terminons par une description des différents fichiers utilisés pour la suite de l'étude des performances. Ces données sont fournies par le système de contrôle.

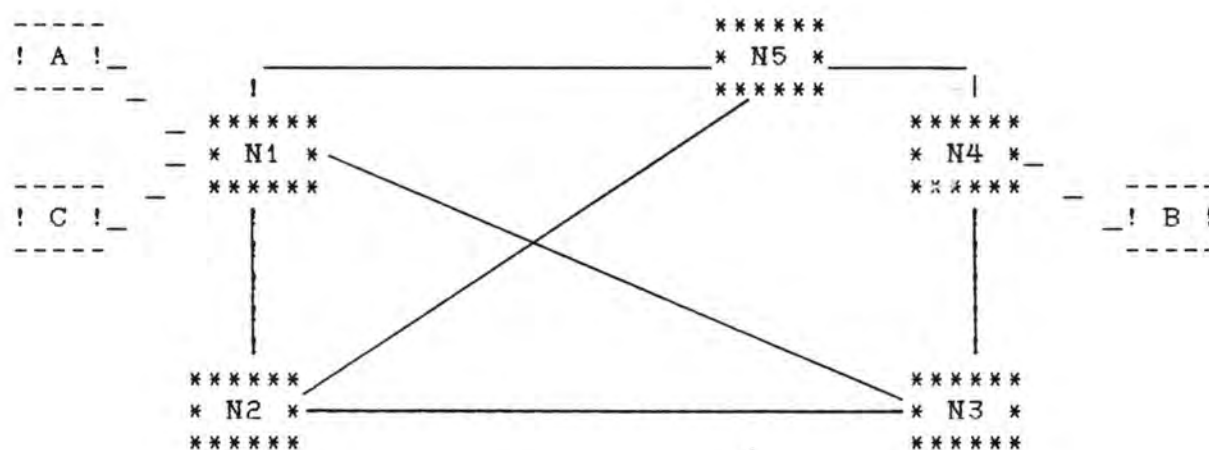
-----

## 1.1 RESEAU A COMMUTATION PAR PAQUETS.

Le lecteur intéressé pourra toujours se référer à l'ouvrage "Transport et traitement de l'information dans les réseaux et systèmes téléinformatiques" écrit par César Macchi et Jean-François Guilbert (Paris, Dunod 1980).

### 1.1.1 Principes.

Rappelons brièvement les principes de la commutation par paquets. Un réseau à commutation par paquets (cfr figure 1.1) est constitué d'un ensemble d'ordinateurs appelés noeuds de commutation (N1, N2, ...) reliés entre eux par des liaisons de données appelées troncs, formant ainsi un réseau maillé.



Légende :

-----	*****
! A !	* N5 *
-----	*****
-----> utilisateurs	-----> noeud
-----	*****

Figure 1.1 : Réseau à commutation par paquets.

Le réseau assure pour ses utilisateurs (A, B, C, ...) le transport de blocs d'informations (contenant les données, une adresse destinataire et un code de contrôle) appelés paquets.

Le transfert d'un paquet entre deux correspondants A et B a lieu comme suit :

A remet le paquet au premier noeud N1 qui le conserve temporairement dans sa mémoire en attendant de pouvoir le passer au noeud suivant.

N1 détermine la liaison suivante du chemin que le paquet va emprunter. Cette opération constitue le routage du paquet, par exemple vers N2.

Dès que N2 pourra l'accepter et que la liaison entre N1 et N2 sera libre, le paquet sera passé à N2, puis aux noeuds suivants (N3 et N4) pour être enfin remis au destinataire B.



Chaque liaison est allouée à la demande lorsque se présente un paquet qui doit l'emprunter. Comme plusieurs paquets peuvent se présenter simultanément pour utiliser une même liaison, il se formera une file d'attente des paquets à transmettre sur cette liaison.

Les paquets arrivant à un noeud sont acheminés indépendamment les uns des autres. Se basant sur le numéro de voie logique contenu dans le paquet, le noeud détermine s'il est le noeud de destination finale ou s'il est intermédiaire. Dans ce cas, il décide où l'envoyer en consultant les tables reprenant toutes les informations sur la topologie du réseau.

### 1.1.2 Accès au réseau à commutation par paquets.

La manière dont un terminal peut accéder aux services d'un réseau à commutation par paquets est définie par un ensemble de protocoles X25. Cet ensemble fournit des conventions aux DTE (Data Terminal Equipment) sur la façon :

- de formater l'information de contrôle et les données dans les paquets
- d'établir, de maintenir et de clôturer un appel
- de gérer la transmission et le flux de données vers et à partir du réseau.

Ces conventions "respectent" les standards pour l'interconnexion de systèmes ouverts adoptés par l'ISO (International Standardisation Organisation). Cet organisme découpe l'architecture d'un réseau en niveaux hiérarchiques et indépendants.

Dans un tel modèle, les entités de chaque niveau communiquent d'après certains protocoles, utilisent les services fournis par les niveaux inférieurs et offrent un service aux niveaux supérieurs (cfr figure 1.2).

Le modèle de référence définit sept niveaux (cfr figure 1.3 page 8) réalisant les fonctions suivantes :

- contrôle physique
- contrôle de la liaison de données
- contrôle du réseau
- contrôle de transport de bout en bout
- niveau session
- niveau présentation
- niveau application

```

                                !paramètres
                                !
                                !Niveau N + 1 !
                                !
                                !paramètres
                                !
                                !Niveau N      !
                                !
                                !paramètres
                                !
                                !Niveau N - 1 !
                                !

```

Figure 1.2 : Découpe en niveaux.

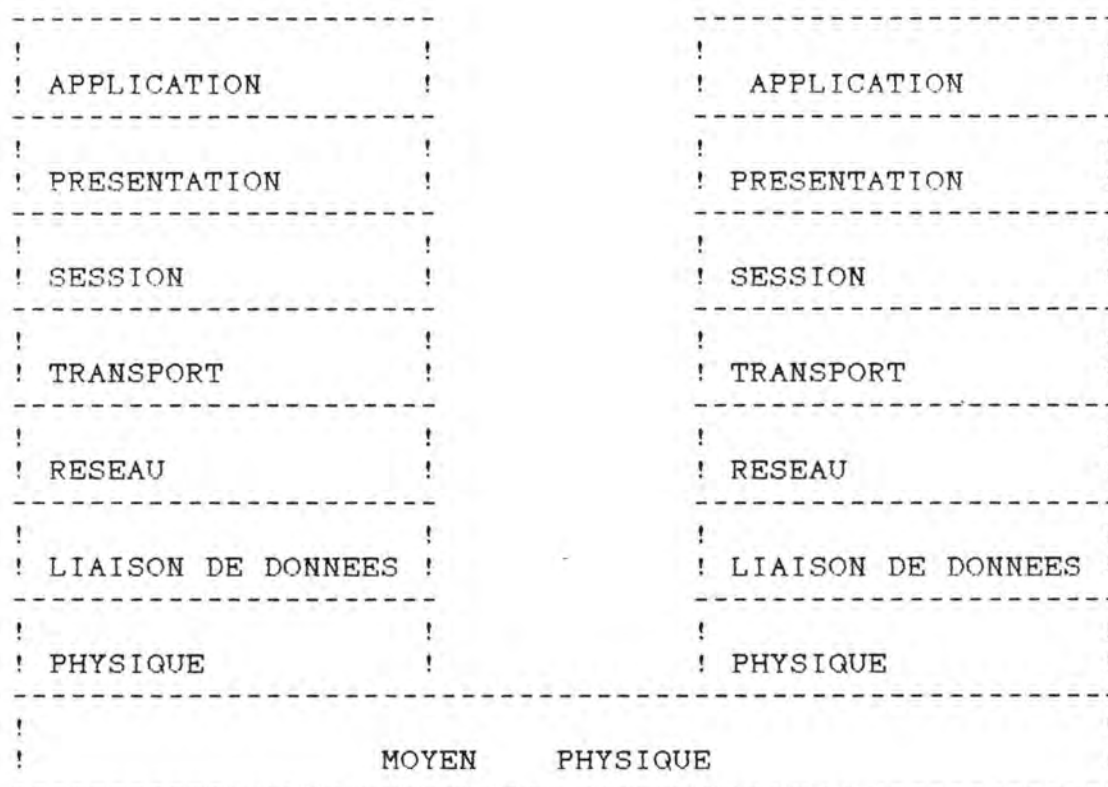


Figure 1.3 : Modèle de référence.

L'interface X25 est constituée des trois derniers niveaux :

- interface de niveau physique
- contrôle de liaison de données
- niveau paquet ou circuit virtuel (service réseau).

a) Niveau physique.

Le niveau physique spécifie l'utilisation d'un circuit point à point duplex qui fournit un chemin de transmission physique entre le terminal et le réseau.

b) Niveau liaison de données.

Le contrôle de liaison de données est compatible avec la procédure de commande de liaison de données à haut niveau (HDLC) standardisée par l'ISO.

Il s'assure que les données sont échangées de façon exacte entre le terminal et le réseau.

L'unité de transfert de ce deuxième niveau est la trame. Elle se compose d'une séquence de délimitation de début de trame, d'un bloc d'informations (avec champs d'adresse et de commande), d'une séquence de contrôle de trame et d'une séquence de délimitation de fin de trame.



Ce deuxième niveau fournit :

- la synchronisation de trames (informations transmises par ce deuxième niveau)
- la transparence
- le transfert d'informations
- le contrôle d'erreur et le recouvrement
- le contrôle de flux
- l'initialisation et la terminaison de la liaison.

### c) Niveau circuit virtuel.

-----

Ce troisième niveau envoie des paquets et supporte le concept de circuit virtuel qui est essentiellement une association logique entre deux DTE.

Il existe deux types de circuits virtuels :

- permanent : association permanente logique sans phase de connexion et de terminaison d'appels.
- commuté : association logique avec procédure d'établissement et de libération de la communication.

Le déroulement d'une communication virtuelle commutée est composé des phases suivantes :

1. L'appelant choisit un numéro libre parmi les numéros de voies logiques dont il dispose. Il envoie alors un paquet de demande d'appel comportant ce numéro, l'adresse de l'abonné demandé, les services du réseau souhaités et un champ de données qu'il peut utiliser par exemple pour insérer un mot de passe. L'appelé reçoit alors un paquet d'indication d'appel, avec un numéro de voie logique choisi parmi les numéros libres à l'instant où l'appel parvient. Et il retourne un paquet d'appel accepté. L'appelant reçoit quant à lui un paquet d'appel connecté, indiquant que le circuit est établi.

2. Les paquets de données sont alors échangés entre les deux utilisateurs.

3. A la fin de l'appel, le circuit peut être déconnecté par un paquet de libération suivi par un paquet de confirmation.

Mais, l'appel entre deux utilisateurs ne peut pas toujours être accepté. Dans ce cas, l'appelant reçoit directement un paquet de libération.

Le niveau circuit virtuel offre les services suivants :

- l'adressage, l'ouverture et la fermeture d'un circuit virtuel
- le transfert de données
- le multiplexage de circuits virtuels sur une seule ligne d'accès
- le contrôle de flux.

### 1.1.3 Avantages des réseaux à commutation par paquets.

Les principaux avantages de tels réseaux sont :

**- le partage des ressources informatiques.**

Les ressources nécessaires à la transmission et en particulier les liaisons ne sont allouées que pour la transmission d'un paquet, et donc facilement partagées entre plusieurs conversations.

**- le routage alternatif des paquets.**

Les noeuds échangent périodiquement des données de routage comme l'utilisation des troncs, la disponibilité des noeuds et des troncs, permettant ainsi de trouver des chemins alternatifs en cas de panne.

Le routage alternatif consiste à choisir un deuxième chemin à travers le réseau lorsque le premier est indisponible. Il exige de ce fait, que chaque noeud soit connecté à au moins deux noeuds adjacents.

**- le multiplexage.**

Plusieurs circuits virtuels peuvent être établis sur un lien physique, permettant ainsi des communications simultanées entre des ordinateurs à travers le réseau. Chaque circuit virtuel est identifié par un numéro de canal logique.

### 1.2 RESEAU SL-10 INSTALLE A LA GENERALE DE BANQUE.

La Générale de Banque installa en 1982 un réseau privé (G-net) à commutation par paquets (X25), pour son application bancaire. Son choix fut guidé par la prise de conscience des éléments suivants :

- volonté d'implémenter un réseau de télécommunication capable de supporter tous les besoins de communication présents et futurs.

- nécessité d'installer un réseau ouvert dont les protocoles d'accès respectent les standards agréés au niveau international par le CCITT (Comité Consultatif International Télégraphique et Téléphonique). Ce type de réseau est indépendant de n'importe quel vendeur et permet une interconnexion plus facile de systèmes multi-vendeurs.

- absence sur le marché d'architectures de communication rencontrant toutes les exigences de la Générale de Banque, lors du choix de son réseau.

- possibilité de connecter les réseaux individuels déjà installés par la Générale de Banque sur le réseau de données unique et par conséquent en réduire les coûts des lignes d'accès.



Le réseau installé par la Générale de Banque est constitué de dix-huit noeuds SL-10 de Northern Telecom (dont deux de tests) interconnectés par des troncs à haute vitesse (allant jusqu'à 64 Kbps). Chaque noeud est caractérisé par un numéro d'identification.

Le nombre de noeuds SL-10 et leur point d'implantation en Belgique ont été établis en fonction du prix du hardware SL-10 et des coûts des lignes employées pour connecter les différents utilisateurs au réseau.

Actuellement, les seize noeuds opérationnels sont installés aux endroits suivants (cfr figures 1.4 et 1.5 page 12) :

* * * * *	* * * * *
* Ville	* Numéro
* * * * *	* * * * *
* Bruges	* 11
* Hasselt	* 12
* Courtrai	* 13
* Tournai	* 23
* Charleroi	* 24
* Herentals	* 30
* Roulers	* 31
* Malines	* 32
* Libramont	* 34
* Bruxelles 1	* 80
* Bruxelles 2	* 81
* Anvers	* 82
* Gand	* 83
* Louvain	* 84
* Liège	* 85
* Mons	* 86
* * * * *	* * * * *

Figure 1.4 : Situation des noeuds SL-10.



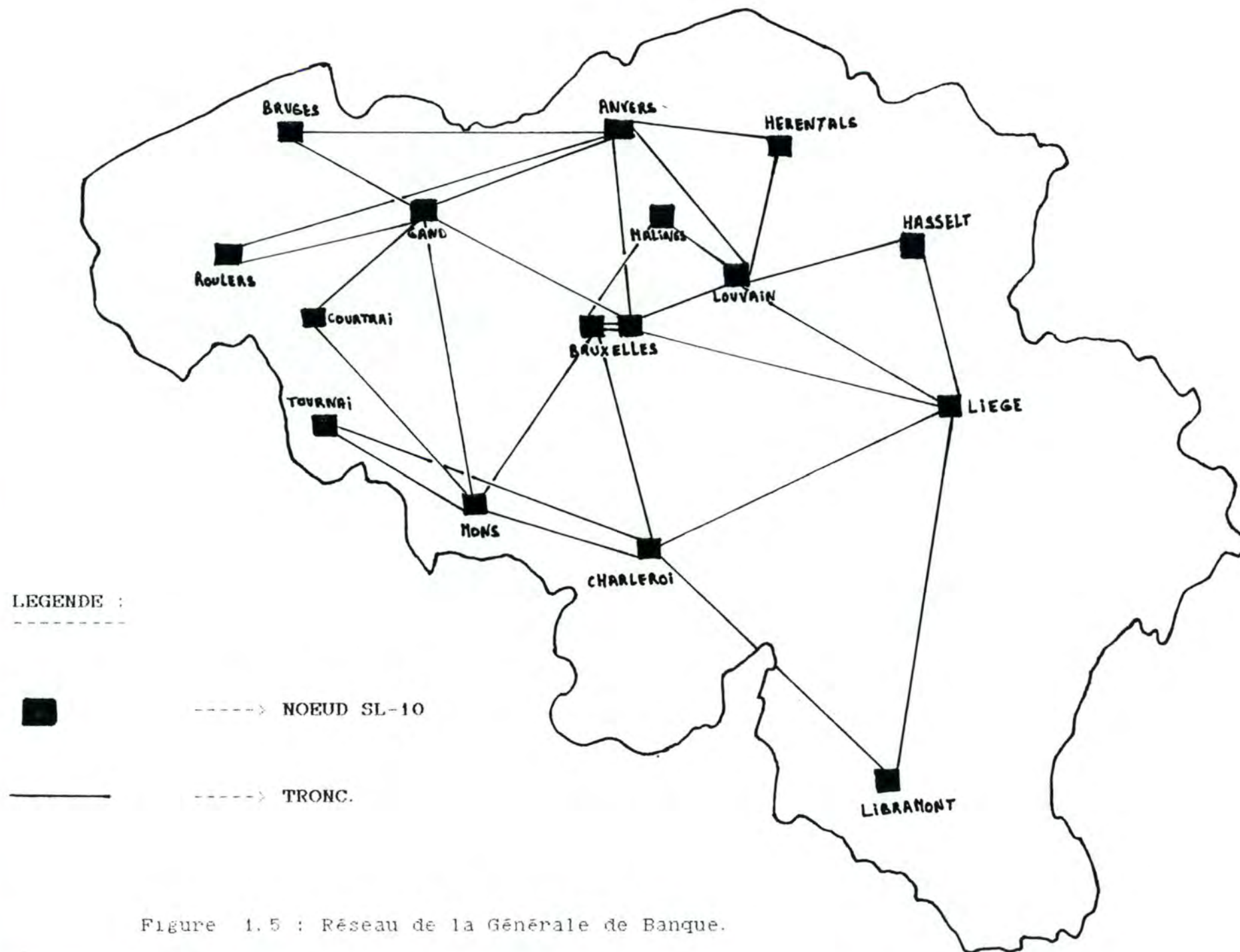


Figure 1.5 : Réseau de la Générale de Banque.

Les utilisateurs accèdent au réseau au moyen de lignes asynchrones (75 à 1200 bps avec une interface PAD) et synchrones X25 (1200 à 9600 bps avec une interface standard et 64 Kbps avec une carte à haute vitesse).

Le réseau offre principalement ses services à trois types d'utilisateurs :

#### \* AGENCES.

-----

Les 1150 agences de la Générale de Banque se répartissent sur toute la Belgique. Elles sont reliées aux noeuds du réseau au moyen de lignes synchrones X25 à 2400 bps. Chaque agence possède un mini-ordinateur équipé d'un disque pour le stockage des différents programmes et d'une base de données locale contenant la situation de ses clients, de terminaux pour la saisie de transactions bancaires et d'imprimantes. Tout ce matériel permet de saisir les informations plus rapidement et plus facilement et permet leur validation en utilisant la base de données 'clients' locale.

Mais une agence ne peut uniquement tenir compte des transactions effectuées chez elle. Toutes les actions bancaires des clients de la Générale de Banque doivent être utilisées pour la mise à jour des bases de données locales. Cette nécessité est remplie par l'interaction avec le groupe d'utilisateurs suivant. Il s'agit des front-ends.

#### \* FRONT-ENDS.

-----

La Générale de Banque dispose de six centres de traitement de données régionaux répartis à travers la Belgique. Chacun d'eux est connecté à un noeud au moyen d'une liaison à haute vitesse de 64 Kbps.

Les noeuds concernés par ces centres sont les suivants :

- Bruxelles (80)
- Anvers (82)
- Gand (83)
- Louvain (84)
- Liège (85)
- Mons (86)

Chaque centre électronique se compose de deux front-ends (un en traitement batch et l'autre en temps réel) et d'un ordinateur host (équipé d'un IBM ou compatible).

Toute agence bancaire dépend d'un et un seul front-end batch (cfr figure 1.6 page 15) avec qui elle communique régulièrement (Il est à noter que les front-ends interactifs ne sont opérationnels que depuis juillet 1986). En vertu de ce type de dépendance, l'ensemble des agences peut être décomposé en six "sous-réseaux", chaque "sous-réseau" étant dirigé par un centre électronique.

Chaque front-end interagit fortement avec les agences qu'il gère. En effet, durant la journée, les agences utilisent les services offerts par le réseau pour expédier vers leur front-end respectif les transactions encodées par les employés et validées sur l'ordinateur de l'agence. Ce transfert se réalise toutes les quinze minutes s'il y a des ordres à envoyer. De toute façon, chaque clôture d'agence donne lieu à une telle procédure.

Après avoir reçu les informations en provenance du front-end, l'ordinateur host commence son travail batch (ce traitement n'a lieu que la nuit). Une communication avec les autres centres électroniques permet au host d'obtenir les informations nécessaires à la consolidation et à la mise à jour des bases de données des différents clients.

Le matin, le réseau est de nouveau sollicité par les front-ends pour mettre à jour les bases de données situées dans les agences. Cette opération se compose des trois phases suivantes :

- chaque front-end appelle ses agences pour signaler que les fichiers de mise à jour sont prêts à être expédiés
- chaque agence appelle son front-end pour signaler qu'elle a besoin de tel fichier entre X et Y heures
- chaque front-end appelle de nouveau ses agences pour envoyer les fichiers de mise à jour demandés.



## Centre électronique

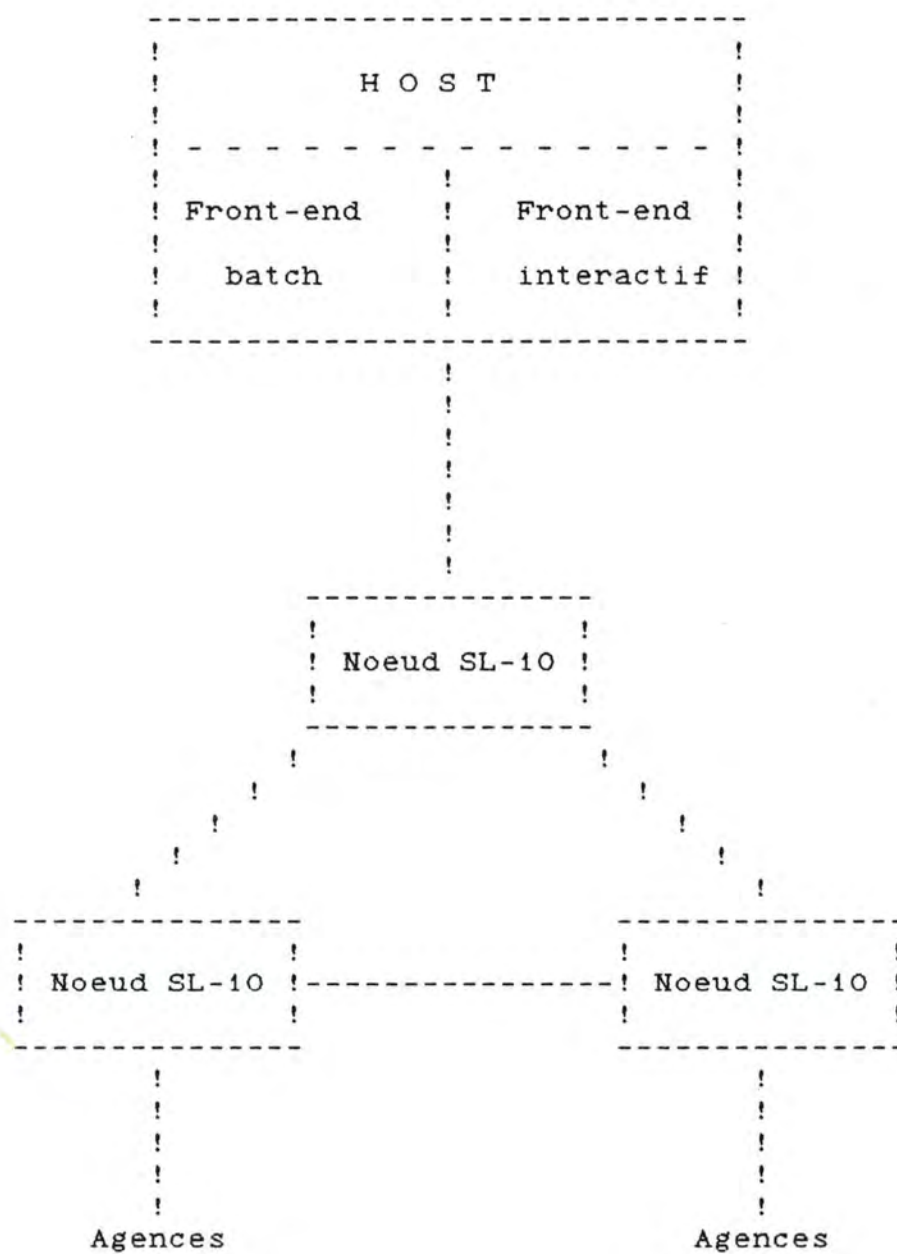


Figure 1.6 : Schéma d'un centre électronique et ses agences.

# \* CALLO1.

-----

Cette catégorie d'utilisateurs peut se décomposer en deux grands groupes caractéristiques :

## - ITI (Interactive Terminal Interface)

Un grand nombre de terminaux (ITI) sont reliés au réseau par des lignes asynchrones. Chaque noeud possède au moins un ITI utilisé comme console pour sortir les alarmes locales, ce qui permet au personnel de contrôle d'être informé immédiatement des problèmes rencontrés au niveau du noeud et de prendre rapidement certaines mesures de correction appropriées.

Comme les terminaux asynchrones ne peuvent recevoir ni émettre des paquets selon le protocole X25, l'installation d'un PAD (Packet Assembly Disassembly) (cfr figure 1.7) s'avère nécessaire. La fonction principale de ce dernier est d'assembler sous forme de paquets les caractères reçus du terminal et inversement de lui délivrer sous forme de caractères les paquets qui lui sont destinés.

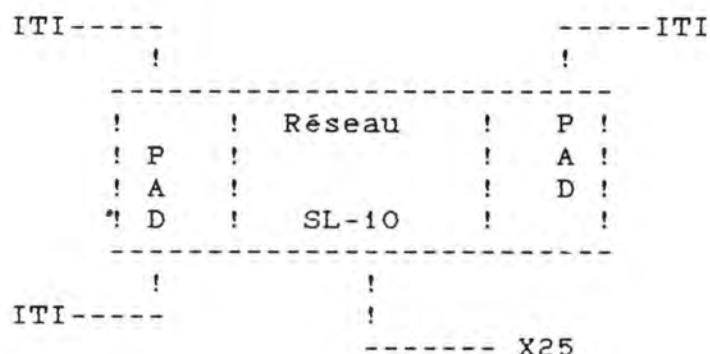


Figure 1.7 : Configuration du PAD.

## - ICI (Internal Call Interface)

Il s'agit d'appels internes au réseau servant de communication entre les processus. Le transit d'alarmes à travers les différents noeuds, le back-up de données statistiques en sont les principales applications.



### 1.3 DESCRIPTION D'UN NOEUD SL-10.

Le système à commutation par paquets de Northern Telecom fournit une communication de données efficace et flexible. Il incorpore un ou plusieurs noeuds SL-10, se composant chacun d'une partie "hardware" et d'une partie "software".

#### 1.3.1 Hardware d'un noeud SL-10.

Le noeud SL-10 est basé sur une architecture multiprocesseur permettant une haute fiabilité et disponibilité du système.

La structure du SL-10 permet la connexion de seize processeurs à un bus commun d'interconnexion. Ces processeurs locaux (cfr figure 1.8) consistent en un ensemble d'unités fonctionnelles appelées processeurs de contrôle, de tronc, de lignes et de mémoire commune.

Les modules de contrôle, de tronc et de lignes communiquent avec le module mémoire commune en utilisant un bus commun d'interconnexion.

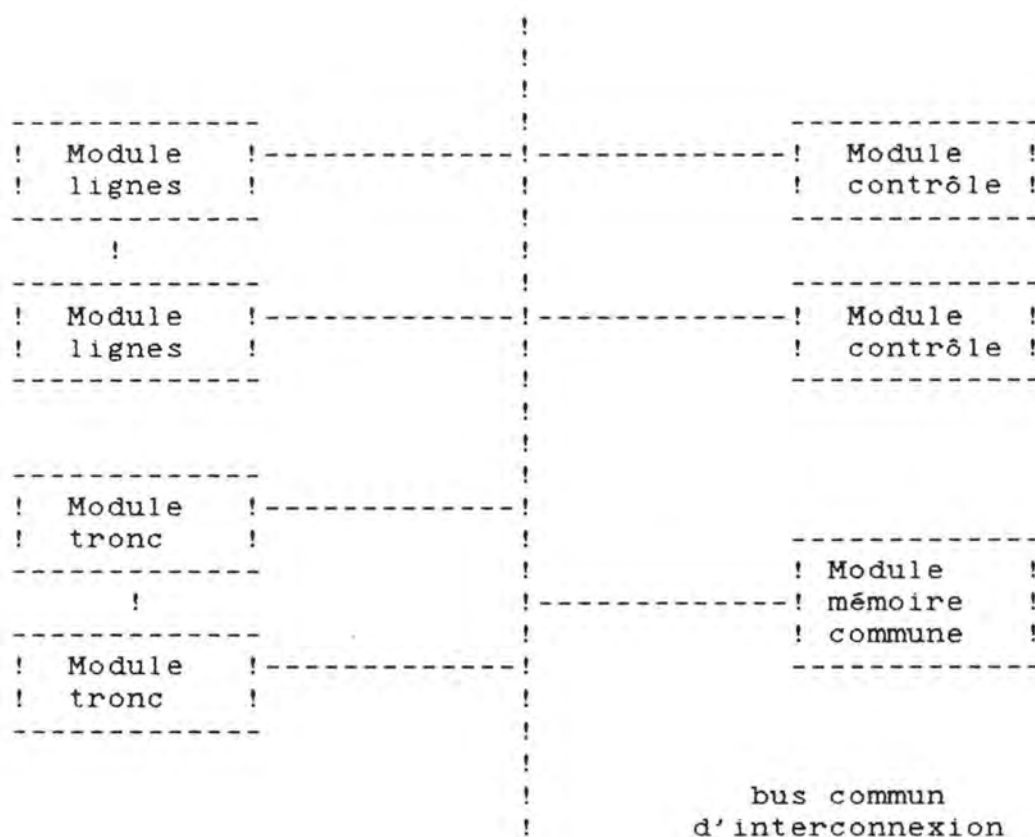


Figure 1.8 : Architecture d'un noeud SL-10.

# \* Module contrôle.

## Fonction :

Fournir au système des ressources de traitement pour établir les circuits virtuels, pour gérer les modifications dans la table de routage, pour changer la configuration du noeud, pour rassembler les données statistiques, de comptabilité et les alarmes. Le processeur de contrôle est également responsable du chargement de tout le logiciel nodal à partir des bandes magnétiques locales.

Composants : - un processeur  
 ----- - une mémoire  
 - un bus switch  
 - une interface bandes magnétiques et/ou disques.

Le "bus switch" fournit une interface entre le module processeur de contrôle et le bus commun d'interconnexion.

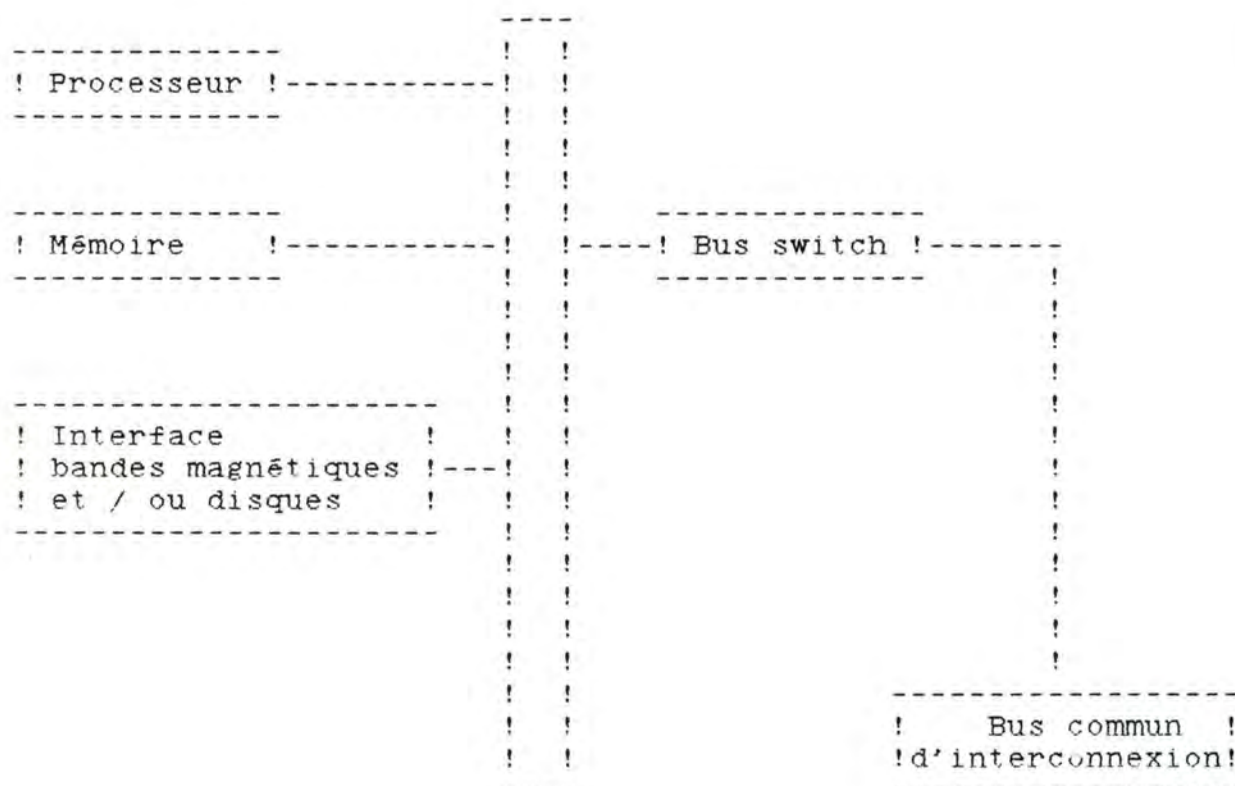


Figure 1.9 : Module processeur de contrôle.

# \* Module tronc.

## Fonction :

Fournir la logique nécessaire pour permettre les connexions à haute vitesse entre les différents noeuds (jusqu'à 64 Kbps).  
Le processeur de tronc procure un service de transmission de données utiles et de contrôle de noeud à noeud et est capable d'acheminer directement les informations reçues vers le processeur de tronc de sortie ou vers le processeur de lignes via le processeur de mémoire commune.

## Composants :

Une armoire processeurs de tronc contient une interface bus et jusqu'à trois processeurs de tronc.

Chaque processeur comprend - un microprocesseur  
- une mémoire ROM  
- une interface tronc.

Le "bus switch" fournit une interface entre le module tronc et le bus commun d'interconnexion.

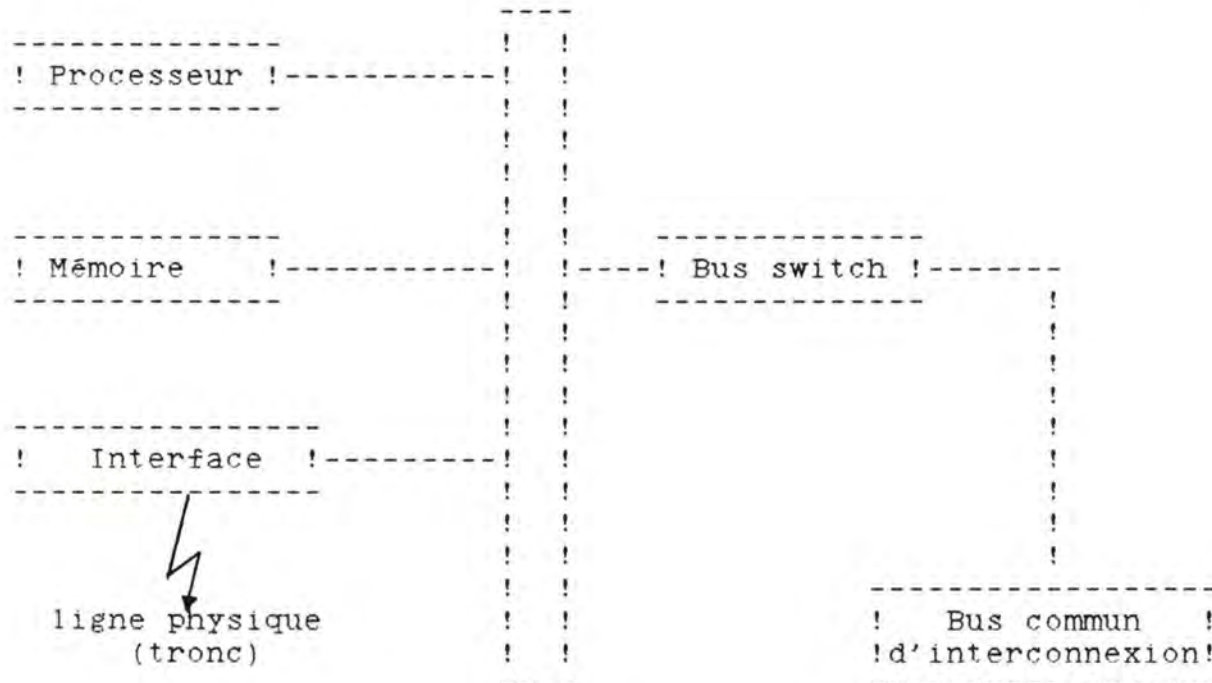


Figure 1.10 : Module tronc.



Les troncs utilisent un protocole particulier qui est différent de X25 et propre à Northern Telecom. Leur unité de transfert est la **trame**.

Une **trame** se compose d'un flag de 21 bits et de 40 slots de 18 bits et peut transporter de 0 à N paquets X25 de données.

Un slot peut renfermer six types d'information :

#### Type 1 :

```

-----
!   !   !   !   !
! 01 ! Noeud de ! Priorité ! LRC !
!   ! destination !   !   !
!   !   !   !   !
-----

```

- Le code "01" signale qu'il s'agit d'un slot de données.
- Le LRC est un contrôle sur les trames.

#### Type 2 :

```

-----
!   !   !
! 01 ! Numéro de référence pour !
!   ! identifier l'appel !
!   !   !
-----

```

- Le code "01" signale qu'il s'agit d'un slot de données.

#### Type 3 :

```

-----
!   !   !   !
! 11 ! Contrôle ! Contrôle !
!   !   !   !
-----

```

- Le code "11" signale qu'il s'agit d'un slot de contrôle pour le tronc.

#### Type 4 :

```

-----
!   !   !   !
! 01 ! 8 bits de ! 8 bits de !
!   ! données ! données !
!   !   !   !
-----

```

- Le code "01" indique qu'il s'agit d'un slot de données.





## \* Module lignes.

- - - - -

### Fonction :

-----

Fournir l'interface nécessaire pour les lignes d'accès connectées à un noeud SL-10.

Le processeur de lignes fournit l'implémentation des différents protocoles d'accès (par exemple X25), du routage des paquets et du contrôle de flux de bout en bout.

### Composants :

-----

- un processeur
- une mémoire locale
- un bus switch
- des interfaces asynchrones pour des lignes allant jusqu'à 1200 bps
- des interfaces synchrones pour des lignes allant jusqu'à 9600 bps
- des interfaces synchrones à haute vitesse pour des lignes allant jusqu'à 64 Kbps (High Speed Line Card).

La mémoire locale stocke le logiciel pour le processeur et fournit un tampon pour les paquets.

Le bus switch permet l'accès au bus commun d'interconnexion lorsqu'un accès à la mémoire commune est exigé.

Un processeur de lignes peut être configuré de trois façons différentes :

- 120 lignes d'accès asynchrones (1200 bps).
- 4 lignes d'accès synchrones à haute vitesse (64 Kbps).
- 2 lignes d'accès synchrones à haute vitesse + 60 lignes d'accès synchrones à 9600 bps.

Les doubles scanners A et B déchargent le processeur de la gestion des interruptions créées pour la réception ou la transmission des caractères de ou vers les cartes de lignes.

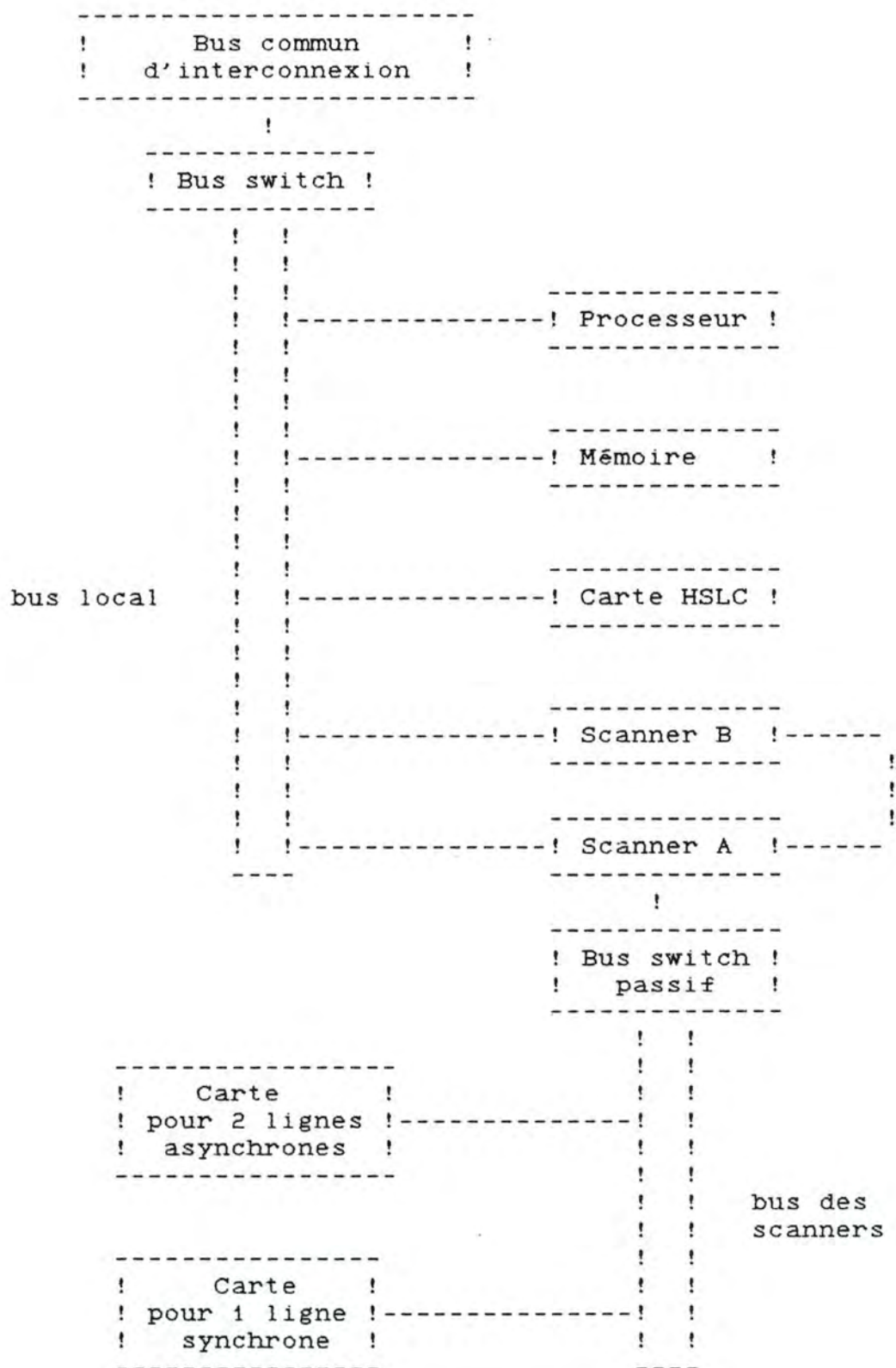


Figure 1.11 : Module processeur de lignes.

# \* Module mémoire commune.

## Fonction :

Fournir des tables de routage, la mémoire de stockage pour les paquets, la gestion des files d'attente, le stockage pour les tables communes et l'arbitrage du bus.  
Il fournit également l'accès au disque si le noeud SL-10 est un noeud de collection de données comptables, statistiques et d'alarmes.

## Composants :

- un bus switch
- un arbitre pour le bus
- une horloge et un gestionnaire des alarmes du noeud
- des processeurs de listes
- une mémoire.

Le bus switch fournit une interface entre le module mémoire commune et le bus commun d'interconnexion.  
L'arbitre du bus contrôle l'accès à ce bus commun d'interconnexion.

La carte horloge et alarmes fournit l'heure du jour, mesure des intervalles de temps et gère les alarmes locales du noeud qui sont affichées sur un panneau de contrôle.

Une paire de processeurs de listes redondants assure quant à elle le service de file d'attente pour les files localisées dans la mémoire commune.

De plus, s'il s'agit d'un noeud collectant les données statistiques, de comptabilité et les alarmes, alors le module possède en plus un disque et un contrôleur de disque.

! Bus commun d'interconnexion !

!

! Bus switch !

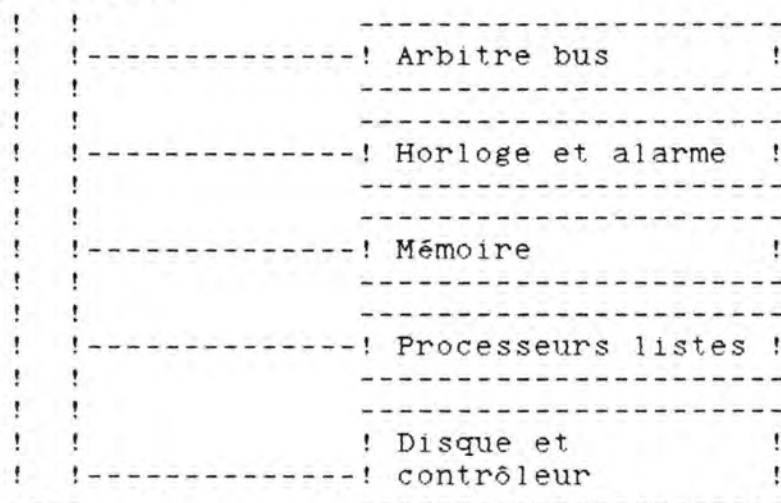


Figure 1.12 : Module mémoire commune



### 1.3.2 Software d'un noeud SL-10.

Le software d'un noeud SL-10 est une conception modulaire, implémentant plusieurs niveaux. Chaque niveau utilise les services fournis par le niveau immédiatement inférieur sans savoir comment celui-ci les réalise et offre ses services au niveau supérieur.

Les différentes fonctions de communication offertes par le réseau SL-10 sont réalisées par quatre niveaux (cfr figure 1.13) :

- noyau du noeud
- niveau sous-réseau
- niveau circuit virtuel
- niveau application.

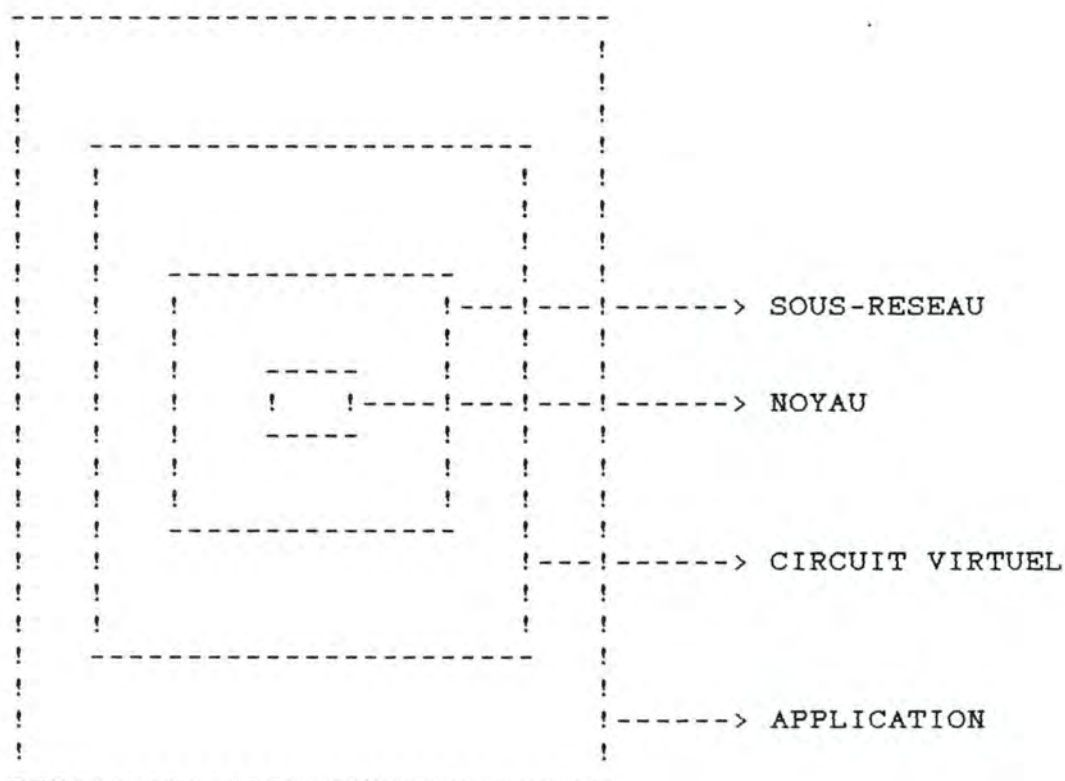


Figure 1.13 : Conception modulaire d'un SL-10.

# \* Noyau.

- - -

Le noyau (le plus bas niveau) inclut le système d'exploitation et fournit les moyens de base nécessaires pour la création de processus et la gestion de messages et un environnement amical au hardware de chaque noeud. Ce niveau permet d'effectuer des changements de hardware sans modifier les autres parties du logiciel.

Tout le travail d'un noeud SL-10 est exécuté par des processus. L'activité d'un processus (cfr figure 1.14) est spécifiée par un programme (exemple : programme de manipulation du niveau liaison de données X25) et par un ensemble de blocs de contrôle spécifique au processus (exemple : valeurs de "time-out" d'une liaison pour le processus servant cette liaison).

Les processus sont activés exclusivement par des messages. Lorsque le système d'exploitation reçoit un message, le processus adéquat est activé et le message lui est passé.

Se basant sur son état passé et sur le contenu du message, le processus peut changer son état et/ou envoyer des messages vers d'autres processus. Il devient alors inactif (il attend un autre message).

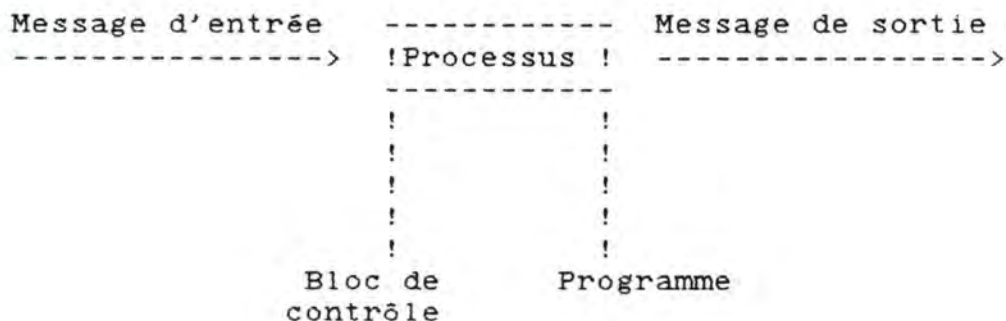


Figure 1.14 : Activité d'un processus.



### \* Niveau sous-réseau.

- - - - -

Ce niveau est responsable du routage de données à travers le réseau. Il incorpore une stratégie d'acheminement pour la communication d'un noeud vers un autre.

Pour aller d'un noeud origine vers un noeud de destination, un paquet X25 traverse un système de troncs SL-10. Chaque système consiste en un minimum de deux chemins internodaux et opère comme un sous-système unique ayant une responsabilité de noeud à noeud pour la livraison correcte de paquets.

Le sous-système utilise un contrôle de redondance cyclique et un protocole avec retransmission pour éviter la perte de paquets.

Ce niveau fournit donc un transfert sans erreur à travers le réseau, mais ne garantit pas la séquentialité et la non-duplication des paquets.

### \* Niveau circuit virtuel.

- - - - -

Ce niveau supporte le flux de paquets contrôlé et bidirectionnel entre le processus servant chaque extrémité d'un circuit virtuel.

Un circuit virtuel est une connexion logique établie entre deux utilisateurs sur laquelle tous les signaux sont transportés sous forme de paquets. L'association peut être temporaire ou permanente.

Un circuit virtuel permet la communication simultanée dans les deux sens et est transparent à l'utilisateur. Il est indépendant de la vitesse des équipements mis en connexion, ce qui permet à des terminaux opérant à des vitesses différentes de communiquer facilement. Les mécanismes de contrôle de flux équilibrent le taux avec lequel les paquets entrent dans le circuit virtuel et le taux avec lequel ils sont délivrés.

Ce niveau garantit la séquentialité, la non-duplication et la conservation intégrale des paquets de données.

### \* Niveau application.

- - - - -

Ce niveau fournit des services à l'utilisateur tels que la prise en charge des protocoles des lignes d'accès et est responsable de l'analyse et du contrôle du réseau.

Il est le seul niveau connu par les gestionnaires. Bien que ses communications avec le réseau soient transparentes à ceux-ci, ses interactions avec ces derniers ne le sont pas.



#### 1.4 CONTROLE DU RESEAU.

La gestion du réseau est assurée par un centre de contrôle, le NCC (Network Control Center) et un centre de collection de données, le DCC (Data Control Center) (cfr figure 1.15).

Le NCC est un software tournant dans les processeurs d'un noeud SL-10 et il s'interface aux gestionnaires à travers des terminaux asynchrones.

Le DCC est un software tournant sur un ordinateur (VAX 750) extérieur au réseau.

Le NCC est utilisé pour surveiller et contrôler le réseau ainsi que pour imprimer sur listing les différents problèmes apparaissant sur celui-ci. Par contre, le DCC permet d'effectuer certaines tâches administratives (génération de rapports), d'entrer des données de service et de stocker le nouveau logiciel dans le réseau.

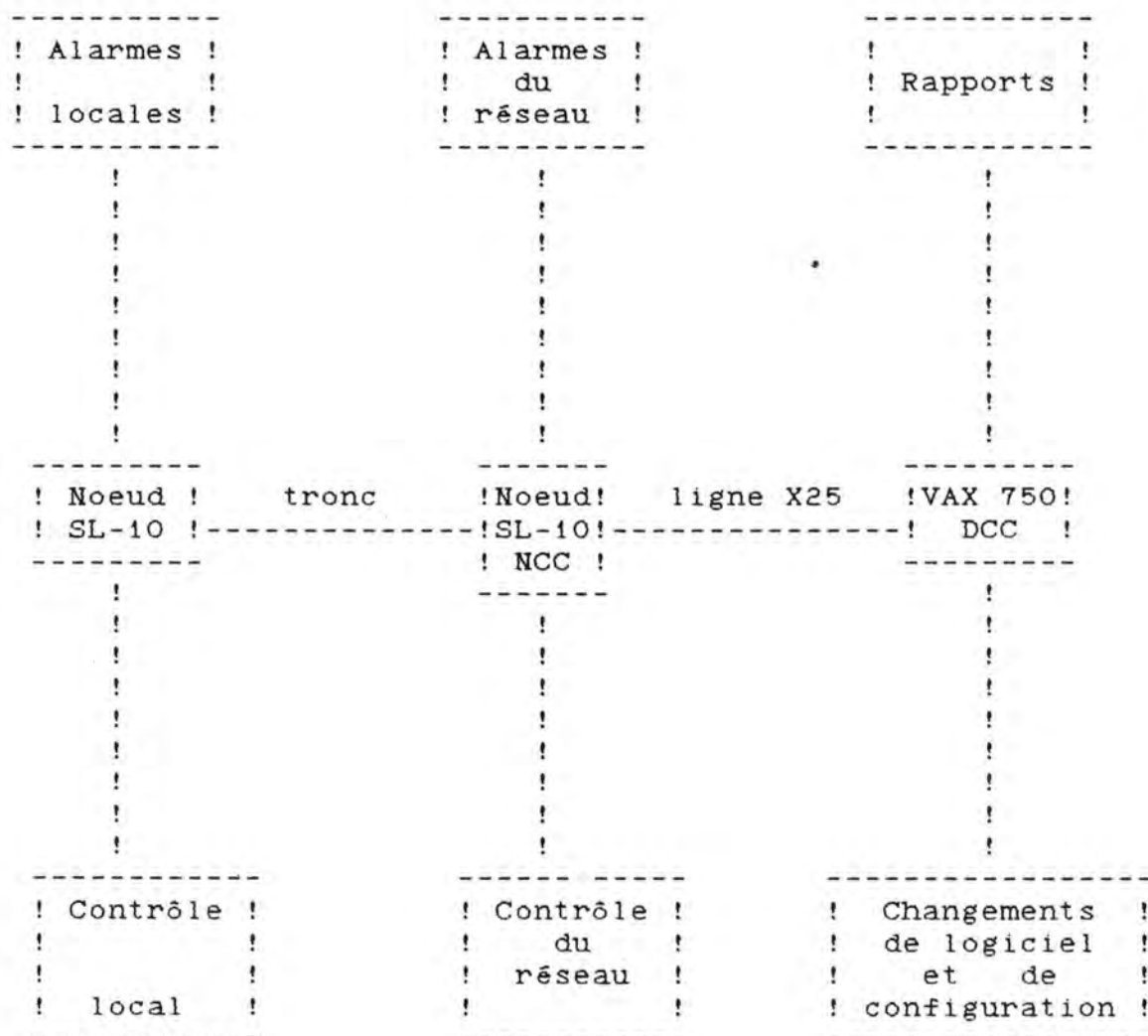


Figure 1.15 : Contrôle du réseau.

#### 1.4.1 Noeud NCC (Network Control Center).

Le NCC est un noeud SL-10 standard (numéroté 80 ou 001), qui est en contact continu avec tous les autres noeuds du réseau.

Sa caractéristique principale est de contrôler leur comportement et leurs activités. Pour réaliser cette tâche, il est doté :

- d'un disque de "spooling" pour le stockage de données (informations de comptabilité, alarmes et statistiques de trafic) en provenance de chaque noeud
- d'une mémoire tampon
- d'un logiciel de gestion de réseau
- d'un nombre considérable de consoles pour le contrôle en temps réel.

Les principales fonctions du NCC sont les suivantes :

1. Chaque noeud du réseau envoie continuellement au NCC des données de performance sur son propre comportement, sur ses troncs et sur ses lignes d'accès connectant les différents utilisateurs.

L'information transmise peut se décomposer en deux catégories :

- détection des erreurs "hardware" et "software"
- statistiques de performance.

Les fautes et les dégradations du réseau parviennent au personnel compétent qui peut immédiatement prendre une action corrective en exécutant des procédures locales ou à distance au moyen d'un terminal.

Les commandes envisagées peuvent :

- exécuter des procédures de test
- activer ou désactiver les composants d'un noeud déterminé
- vérifier des compteurs de statistiques
- contrôler des données de service.





#### 1.4.2 DCC (Data Control Center).

Le DCC est un ordinateur VAX 11/750 relié au NCC (Network Control Center) au moyen d'un interface X25, comme n'importe quel autre utilisateur. Leur communication nécessite l'établissement d'un circuit virtuel commuté car il n'y a pas de liaison permanente entre eux.

Son but est double :

-----

\* Dans le sens DCC ---> réseau.

- - - - -

Le DCC permet :

- de mettre à jour les logiciels utilisés dans les noeuds du réseau
- d'implémenter de nouveaux logiciels dans les différents noeuds
- de modifier les données de service nécessaires au fonctionnement de chaque noeud.

Pour réaliser les différentes modifications et le chargement des nouvelles données, le NCC établit un circuit virtuel commuté avec le DCC.

Le NCC peut alors recevoir les informations de mise à jour provenant du DCC et les expédier à son tour vers les différents noeuds du réseau afin qu'elles soient stockées sur bandes magnétiques ou disques locaux. Grâce à une commande du NCC, les données sont chargées à l'intérieur de chaque noeud et deviennent par la même occasion opérationnelles.

\* Dans le sens réseau ---> DCC.

-----

Dans ce cas, le DCC est considéré comme un centre de collection et d'administration pour l'archivage, le triage et l'agencement des données collectées par le NCC, en un ensemble cohérent.

Par manque de mémoire secondaire suffisante, le NCC (Network Control Center) n'est pas capable de stocker de façon permanente les différentes données (de comptabilité, statistiques et alarmes) en provenance des autres noeuds du réseau. Il est donc obligé de les envoyer vers le DCC (cfr figure 1.17 page 33). De ce fait, il utilise les services de "dumping" offerts par ce dernier.

Le processus de "dump" peut être décomposé en plusieurs phases :

- Périodiquement (toutes les vingt minutes), le NCC fait une demande de connexion à l'ordinateur DCC. Un circuit virtuel devient alors disponible pour la communication. L'établissement de ce circuit constitue un appel interne au réseau (ICI).
- Le NCC se connecte au DCC.
- Le NCC commence le transfert de données en envoyant une commande "START DUMP".
- Le "dump" des données se réalise secteur par secteur. Chaque secteur donne lieu à un acquit de réception de la part du DCC. En cas de problème, le "dump" est arrêté et un autre essai a lieu vingt minutes plus tard.
- Lorsque le transfert de données est terminé, le NCC envoie la commande "END OF DUMP" au DCC.
- A la suite d'une réception totale et réussie de données, le DCC signale "OK TO DELETE" et la déconnexion s'ensuit automatiquement.

L'exécution de ce processus a pour effet de copier les données brutes collectées par le NCC sur deux fichiers différents du DCC "YMMDDHHMM.RD1" et "YMMDDHHMM.RD2" (backup du .RD1) où

- "Y" est le dernier chiffre de l'année courante
- "MM" est le mois courant
- "DD" est le jour courant
- "HH" est l'heure
- "MM" sont les minutes.

Les noms de ces deux fichiers sont respectivement ajoutés aux fichiers "DATA LIST.RD1" et "DATA LIST.RD2".

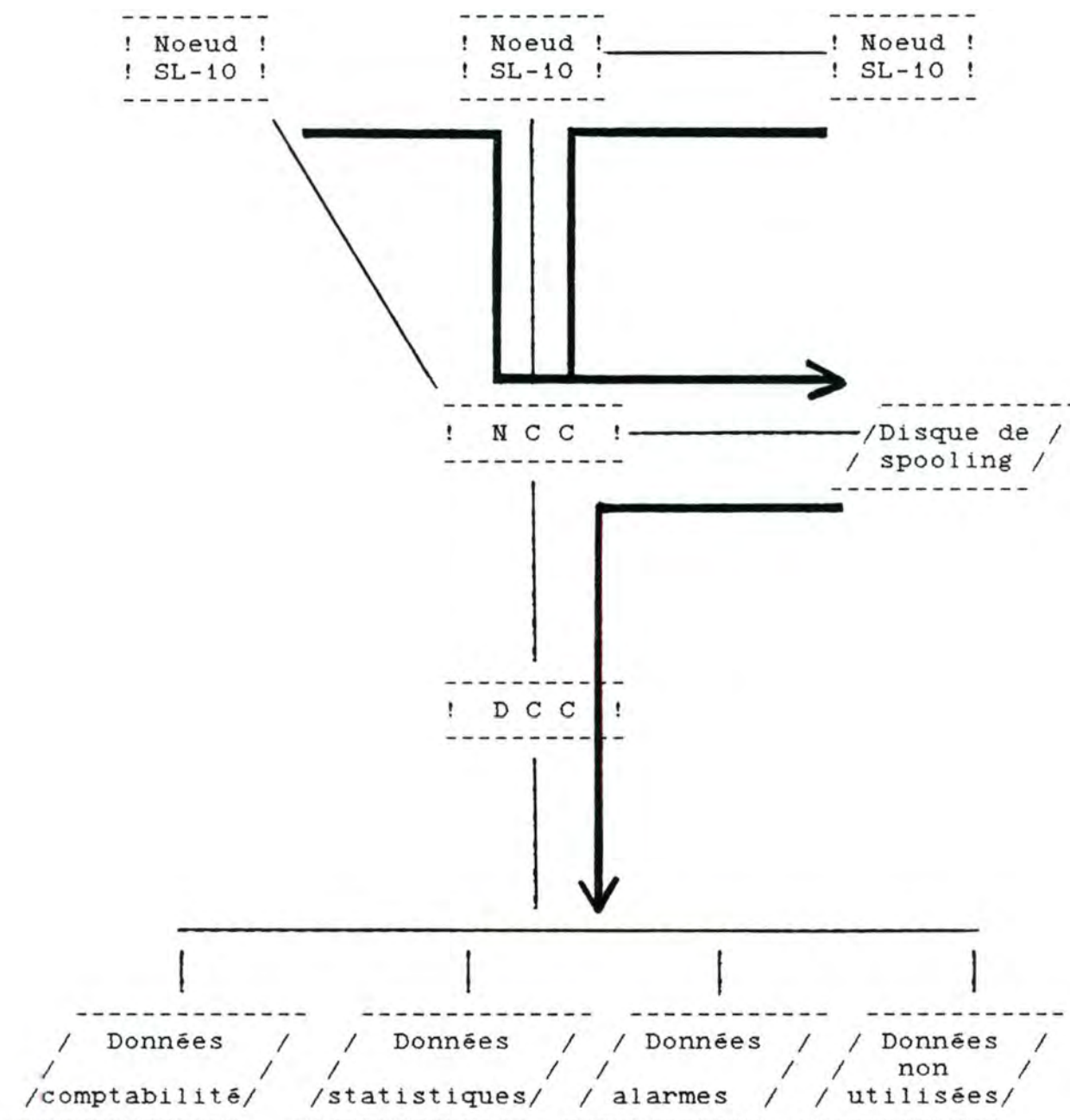


Figure 1.17 : Flux de données NCC - DCC.



En fin de journée, on exécute un programme "DAILY" qui traite les fichiers de données brutes dont le nom est enregistré dans "DATALIST.RD1". Il se compose de plusieurs procédures ayant des fonctions spécifiques.

#### \* Procédure "EFEDIT".

-----

But : Diviser les fichiers de données brutes en :

---

- trois fichiers formatés

"YMMDDACNT.DAT" --> enregistrements de comptabilité

"YMMDDSTAT.DAT" --> enregistrements statistiques

"YMMDDALRM.DAT" --> enregistrements alarmes

- un fichier non formaté

"YMMDDUNUS.DAT" --> information non utilisée.

où "Y" représente le dernier chiffre de l'année

"MM" représente le mois courant

"DD" représente le jour où les données ont été collectées.

Les données du fichier "YMMDDACNT.DAT" sont collectées par les noeuds source et destination lorsqu'un appel entre deux utilisateurs est actif. Elles sont expédiées vers le noeud NCC à la fin de chaque appel ou périodiquement si l'appel dépasse une certaine durée.

Elles consistent en - date et heure du début et de fin d'appel  
 - nombre de paquets envoyés et reçus  
 - raison de la terminaison de l'appel  
 - adresse réseau des utilisateurs  
 - paramètres de comptabilité.

Les données du fichier "YMMDDSTAT.DAT" sont des informations statistiques. En effet, le NCC exige que chaque noeud du réseau expédie des informations à des intervalles de temps donnés (toutes les quinze minutes). Il s'agit de renseignements concernant la disponibilité des composants du réseau, leurs charges moyennes et maximum et le nombre de paquets reçus et transmis.

Les données du fichier "YMMDDALRM.DAT" contiennent les alarmes. Une alarme est générée lorsqu'un problème "hardware" ou "software" est détecté ou résolu. L'information sur les conditions de toute erreur est envoyée simultanément vers la console locale et vers le noeud NCC.

# \* Procédure "ASDAAC".

-----

But : Changer le nom des trois fichiers formatés par la procédure  
 --- "EFEDIT".

```
"YMMDDACNT. DAT" ---> "ACNT. DAT"
"YMMDDSTAT. DAT" ---> "STATIS. DAT"
"YMMDDALRM. DAT" ---> "ALARMS. DAT"
```

# \* Procédure "ASSFSP".

-----

Buts : 1. Séparer le fichier de statistiques "STATIS. DAT" en  
 ---- trois fichiers individuels pour chaque type de  
 statistiques.

- "LINEPROC. DAT" : statistiques sur les processeurs de  
lignes de chaque noeud
- "TRUNK. DAT" : statistiques sur les troncs du réseau
- "SUBLINE. DAT" : statistiques sur les lignes X25  
reliant les différents utilisateurs  
au réseau.

2. Trier sur le numéro du noeud, la date et l'heure les  
trois fichiers précédents :

```
- "LINEPROC. DAT" ---> "LINEPROC. SRT"
- "TRUNK. DAT" ---> "TRUNK. SRT"
- "SUBLINE. DAT" ---> "SUBLINE. SRT"
```

## 1.5 DESCRIPTION DES FICHIERS FOURNIS PAR LA PROCEDURE 'DAILY'.

Nous ne reprenons, dans cette partie, que les différentes données des fichiers nécessaires pour la suite de l'étude. Nous ne présentons pas le fichier de statistiques des lignes X25 car nous n'avons pas utilisé les zones qu'il renferme. Pour une description détaillée de tous les fichiers, nous renvoyons le lecteur à l'annexe page 1.

### 1.5.1 Fichier de comptabilité des appels : 'ACNT.DAT'.

Ce fichier comptabilise tous les appels entre les utilisateurs du réseau.

Chaque utilisateur possède une adresse réseau ayant la forme suivante "NNXYZZZ" où

- "NN" représente le numéro du noeud sur lequel est relié l'utilisateur
- "XX" est toujours égal à "00"
- "Y" représente le type de ligne employée pour relier l'utilisateur au réseau.  
Cette zone vaut :

- "0" pour un appel interne
- "1" pour une ligne asynchrone
- "2" pour une ligne synchrone

- "ZZZ" représente le numéro attribué à la ligne reliant l'utilisateur au réseau.

Lorsqu'un appel réussit entre deux utilisateurs du réseau (nommés respectivement appelant et appelé), chaque noeud concerné par l'appel enregistre des informations (ceci signifie donc qu'à un appel correspondent deux enregistrements dans le fichier, duals l'un de l'autre).

S'il ne réussit pas, un seul enregistrement est comptabilisé par le noeud supportant l'appelant.

#### **\* Format des enregistrements du fichier 'ACNT.DAT'.**

##### **A) OFFICE\_CODE : PIC X(3).**

-----  
Cette zone représente le numéro d'identification du noeud générant l'enregistrement courant

##### **B) DATE : PIC X(6).**

----  
Cette zone représente la date à laquelle l'appel concerné par l'enregistrement a débuté.

Elle est stockée sous la forme "YYMMDD" où

- "YY" représente les deux derniers chiffres de l'année
- "MM" représente le mois
- "DD" représente le jour.



C) **TIME** : PIC X(6).

-----

Cette zone représente l'heure de fin de l'enregistrement.  
Elle est stockée sous la forme "HHMMSS" où

- "HH" représente l'heure
- "MM" représente les minutes
- "SS" représente les secondes.

D) **RECORD\_TYPE** : PIC X(3).

-----

Cette zone contient une information décimale représentant le mot binaire suivant :

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
!	!	!	!	!	!	!	!
Identification de l'enregistrement				A	B	C	D

- Identification de l'enregistrement : 3 pour la comptabilité
- A : 0 pour le format de l'enregistrement national.  
1 pour le format de l'enregistrement international.  
Cette zone vaut toujours "0" à la Générale de Banque.
- B : zone non utilisée.
- C , D : Si C = 0 et D = 0, alors il s'agit d'un enregistrement intermédiaire.  
Si C = 0 et D = 1, alors il s'agit du premier enregistrement de l'appel.  
Si C = 1 et D = 0, alors il s'agit du dernier enregistrement de l'appel.  
Si C = 1 et D = 1, alors il s'agit de l'unique enregistrement de l'appel.

E) **LOCAL\_DNA** : PIC X(10).

-----

Cette zone représente l'adresse réseau de l'utilisateur local de l'appel (c'est-à-dire l'endroit où l'enregistrement est généré). Cette adresse apparaît comme étant l'adresse de l'utilisateur éloigné dans l'enregistrement généré par le noeud à l'autre bout de l'appel.

F) **REMOTE\_DNA** : PIC X(10).

-----

Cette zone représente l'adresse réseau de l'utilisateur éloigné de l'appel. Cette adresse apparaît comme étant l'adresse de l'utilisateur local dans l'enregistrement généré par le noeud à l'autre bout de l'appel.

G) **SEGMENTS\_RECVD** : PIC X(8).

-----  
 Cette zone contient le nombre de segments reçus par l'utilisateur local. Elle comprend les données, les interruptions et la confirmation de ces interruptions. Un segment est une unité de transfert ayant une longueur de 64 ou 128 bytes. Les paquets dont la longueur est inférieure à celle d'un segment interviennent pour un segment.

H) **SEGMENTS\_SENT** : PIC X(8).

-----  
 Cette zone contient le nombre de segments envoyés par l'utilisateur local. Les règles sont les mêmes que pour la zone précédente.

I) **STIME** : PIC X(6).

-----  
 Cette zone représente l'heure de début de l'enregistrement. Elle est stockée sous la forme "HHMMSS" où

- "HH" représente l'heure
- "MM" représente les minutes
- "SS" représente les secondes.

J) **CLEAR\_CODE** : PIC X(2).

-----  
 Cette zone contient un code permettant d'identifier la cause de la terminaison d'un appel.

K) **CALL\_PARAMETERS** : PIC X(2).

-----  
 Cette zone contient une information décimale représentant le mot binaire suivant :

bit 3	bit 2	bit 1	bit 0
!	!	!	!

-----

où - le bit 0 vaut 0 si celui qui appelle paie la communication  
       1 si celui qui est appelé paie.

- le bit 1 vaut 0 si l'utilisateur local est l'appelant  
       1 si l'utilisateur local est l'appelé.

- le bit 2 vaut 0 si la priorité de l'envoi est normale  
       1 si la priorité de l'envoi est grande.

- le bit 3 vaut 0 si un circuit virtuel commuté est employé  
       1 si un circuit virtuel permanent est utilisé.

## Exemples :

- - - - -  
 a) Si la zone 'CALL\_PARAMETERS' vaut "00", alors la forme binaire est égale à "0000". Ce qui signifie que :

- l'utilisateur local est appelant
- l'utilisateur local paie la communication
- les segments sont envoyés avec une priorité normale
- le circuit virtuel utilisé est commuté.

b) Si la zone 'CALL\_PARAMETERS' vaut "06", alors la forme binaire est égale à "0110". Ce qui signifie que :

- l'utilisateur local est appelé
- l'utilisateur éloigné paie la communication
- les segments sont envoyés avec une haute priorité
- le circuit virtuel utilisé est commuté.

L) **CALL\_DURATION** : PIC X(10).

-----  
 Cette zone est la différence entre l'heure de fin (TIME) et l'heure de début (STIME) de l'enregistrement. La durée est exprimée en secondes.



### 1.5.2 Fichier des statistiques des processeurs de lignes : 'LINEPROC. SRT'.

Rappelons qu'un processeur de lignes fournit l'interface nécessaire pour les lignes d'accès connectées à un noeud SL-10. Chaque processeur est identifié par un numéro interne au noeud qui le contient.

Chaque enregistrement de ce fichier représente les informations pour un processeur de lignes pendant une période statistique de quinze minutes. Ces informations sont obtenues normalement par quinze échantillons d'une minute chacun.

#### \* Format des enregistrements du fichier 'LINEPROC. SRT'.

##### A) OFFICE\_CODE : PIC X(3).

-----  
Cette zone représente le numéro du noeud supportant le processeur de lignes concerné par l'enregistrement courant.

##### B) DATE : PIC X(6).

-----  
Cette zone représente la date de génération de l'enregistrement.

Elle est stockée sous la forme "YYMMDD" où

- "YY" représente les deux derniers chiffres de l'année
- "MM" représente le mois courant
- "DD" représente le jour courant.

##### C) TIME : PIC X(6).

-----  
Cette zone représente l'heure de génération de l'enregistrement.

Elle est stockée sous la forme "HHMMSS" où

- "HH" représente l'heure
- "MM" représente les minutes
- "SS" représente les secondes.

##### D) LP\_NUMBER : PIC X(3).

-----  
Cette zone détermine le numéro du processeur de lignes concerné par l'enregistrement.

##### E) LP\_LOAD\_FREE\_QUEUE : PIC X(8).

-----  
Cette zone représente le nombre total de blocs libres mis à la disposition du processeur de lignes lors de l'initialisation.

##### F) LP\_AVAIL\_FREE\_QUEUE : PIC X(8).

-----  
Cette zone représente le nombre total de blocs disponibles observés à chaque échantillon d'une minute.

Pour expliquer les zones suivantes de l'enregistrement, certaines notions relatives au travail d'un processeur de lignes sont indispensables.

Continuellement, le processeur de lignes boucle dans le gestionnaire de processeur (appelé LP\_DRIVER) pour vérifier s'il y a quelque chose à faire. Lorsqu'il trouve du travail, le processus adéquat pour le genre d'application demandé est déclenché.

Lorsque le processus se termine, le processeur de lignes recommence à boucler dans le gestionnaire de processeur.

G) **DR\_RUN\_COUNT** : PIC X(10).

-----

Cette zone représente le nombre de fois que le processeur de lignes a vérifié durant la période statistique, s'il y avait quelque chose à traiter.

H) **DR\_RUN\_TIME** : PIC X(10).

-----

Cette zone représente le temps total passé par le processeur de lignes à vérifier s'il y avait quelque chose à faire (il s'agit du temps passé dans le LP\_DRIVER).

I) **OS\_RUN\_COUNT** : PIC X(10).

-----

Cette zone représente le nombre de fois que le processeur a trouvé du travail durant la période statistique.

J) **OS\_RUN\_TIME** : PIC X(10).

-----

Cette zone représente le temps total mis par le processeur de lignes pour traiter le travail trouvé durant la période statistique.

### 1.5.3 Fichier des statistiques de tronc : 'TRUNK.SRT'.

Rappelons qu'un tronc est un chemin entre deux noeuds SL-10 du réseau.

Chaque extrémité d'un tronc enregistre les informations statistiques relatives à celui-ci.

Chaque tronc d'un noeud déterminé possède un numéro d'identification interne.

Chaque enregistrement de ce fichier représente les informations pour un tronc pendant une période statistique de quinze minutes.

Pour la description des enregistrements, nous utilisons la terminologie suivante :

- noeud local : noeud concerné par l'enregistrement courant.
- noeud éloigné : noeud relié au noeud local au moyen du tronc faisant l'objet de l'enregistrement courant.
- numéro du tronc local : numéro du tronc concerné par l'enregistrement. Ce numéro est interne au noeud local.
- numéro du tronc éloigné : numéro du tronc concerné par l'enregistrement. Ce numéro est interne au noeud éloigné.

#### \* Format des enregistrements du fichier 'TRUNK.SRT'.

##### A) OFFICE\_CODE : PIC X(3).

-----  
 Cette zone représente le numéro du noeud local enregistrant les statistiques du tronc.

##### B) DATE : PIC X(6).

----  
 Cette zone représente la date de génération de l'enregistrement.

Elle est stockée sous la forme "YYMMDD" où

- "YY" représente les deux derniers chiffres de l'année
- "MM" représente le mois courant
- "DD" représente le jour courant.

##### C) TIME : PIC X(6).

----  
 Cette zone représente l'heure de génération de l'enregistrement.

Elle est stockée sous la forme "HHMMSS" où

- "HH" représente l'heure
- "MM" représente les minutes
- "SS" représente les secondes.

##### D) LOCAL\_TRUNK\_NO : PIC X(3).

-----  
 Cette zone correspond au numéro du tronc local concerné par l'enregistrement.



E) **REMOTE\_NODE\_NO** : PIC X(3).

-----  
Cette zone représente le numéro du noeud éloigné concerné par l'enregistrement.

F) **REMOTE\_TRUNK\_NO** : PIC X(3).

-----  
Cette zone correspond au numéro du tronc éloigné concerné par l'enregistrement.

G) **IDLE\_SLOTS** : PIC X(5).

-----  
Cette zone représente le nombre total de "slots" libres divisé par 256, reçus durant la période statistique, par le noeud local à travers le tronc concerné par l'enregistrement.

H) **FRAMES\_RECVD** : PIC X(8).

-----  
Cette zone représente le nombre total de trames reçues (au sens Northern Telecom) par le noeud local durant la période statistique.

## CHAPITRE 2

### PRESENTATION DES VARIABLES A ETUDIER.

-----

#### Résumé :

L'objectif de ce chapitre est de présenter les différentes variables à mesurer afin d'évaluer la performance du réseau, ainsi que les variables susceptibles d'influencer leur comportement. Pour la suite de l'étude, nous utiliserons un vocabulaire propre à l'analyse des données. Nous appellerons "**variables à expliquer**" les mesures de performance et "**variables explicatives**" les autres.

Pour le dernier type de données, nous donnons quelques résultats statiques, indépendants de tout trafic journalier causé par les utilisateurs du réseau.

-----

## 2.1 VARIABLES A EXPLIQUER.

Les variables à expliquer ont été définies par les gestionnaires du réseau. Certaines d'entre elles se déduisent de quelques zones appartenant aux fichiers produits par la procédure 'DAILY' et au fichier de comptabilité des appels 'ACNT.DAT'. Les autres nécessitent, quant à elles l'utilisation d'informations supplémentaires, recueillies en temps réel.

### 2.1.1 Pourcentage d'utilisation des processeurs de lignes : TAUXPL.

Il s'agit du temps d'utilisation d'un processeur de lignes durant une période divisé par le temps total de la période à savoir plus ou moins quinze minutes.

### 2.1.2 Pourcentage de disponibilité de la mémoire des processeurs de lignes : TAUXMEM.

Nous expliquons la manière de calculer ce pourcentage grâce à l'exemple suivant :

Exemple : Lors de son initialisation, un processeur de lignes dispose d'un bloc de mémoire de 200 bytes. Durant une période de quinze minutes, on réalise cinq échantillonnages et on mesure la taille du bloc encore disponible.

* * * * *										* * * * *									
*            Numéro de            *										*            Taille du bloc            *									
* l'échantillonnage *																			
* * * * *										* * * * *									
*																			

Le pourcentage de disponibilité de la mémoire du processeur est donc égal à  $(190 + 180 + 181 + 187 + 192) / (200 * 5)$ , c'est-à-dire 93 %.

### 2.1.3 Pourcentage d'utilisation des troncs : TAUXTR.

Il s'agit du rapport entre le nombre de slots (unités de transport de Northern Telecom) contenant de l'information utile (données et caractères de contrôle) et le nombre de slots envoyés durant une période de plus ou moins quinze minutes.

Il est à noter que des trames sont constamment envoyées sur les troncs, même si aucune information utile ne s'y trouve.

### 2.1.4 Temps d'existence d'un appel : TEMPS.

Il s'agit du temps d'existence d'une liaison (circuit virtuel) entre deux utilisateurs du réseau.





### 2.1.8 Pourcentage d'utilisation des processeurs de contrôle : TAUXPC.

Il s'agit du temps d'utilisation d'un processeur de contrôle durant une période divisé par le temps total de la période à savoir plus ou moins quinze minutes.

Aucune information fournie par les fichiers mis à notre disposition ne permet de calculer ce pourcentage d'utilisation.

### 2.1.9 Temps moyen entre une demande d'appel et sa confirmation : TEMPSAP.

Il s'agit du temps moyen nécessaire pour établir une liaison entre deux utilisateurs. Ce temps représente la différence entre le moment où un utilisateur fait explicitement une demande d'appel et le moment où cet appel est confirmé par le destinataire.

Aucune information fournie par les différents fichiers disponibles ne permet d'estimer ce temps moyen.

Pour les deux dernières variables à expliquer, il est nécessaire de développer de nouveaux outils dont le but est d'effectuer en temps réel les différentes mesures de performance. Il est à noter que ce travail ne fait pas partie du présent mémoire ; il peut être vu comme un prolongement de celui-ci.

## 2.2 VARIABLES EXPLICATIVES DES MESURES EFFECTUEES SUR LES NOEUDS.

Les variables explicatives ont été quant à elles déterminées par tâtonnement au vu du fonctionnement du réseau. Il s'agit de toutes les données, qui étaient selon nous, susceptibles d'influencer les variables de performance.

### 2.2.1 Type d'application : AP.

En fonction de la période dans la journée, il est possible de déterminer le type principal d'application utilisant les services offerts par le réseau.

Ainsi par exemple, le jour, les agences envoient régulièrement les ordres enregistrés par les employés. A chaque clôture d'agence, celle-ci force l'envoi des informations vers les front-ends qui les expédient vers l'ordinateur host. Le traitement de ces données se déroule uniquement la nuit.


La mise à jour des bases de données situées dans les agences se fait en fin de nuit, lorsque le front-end a signalé la terminaison de son travail. Nous sommes donc dans cet exemple, en présence de deux types d'application : celui des agences la journée et celui des front-ends la nuit.

### 2.2.2 Vitesse des troncs : VIT.

Cette variable enregistre les différentes vitesses des troncs utilisés par le réseau de la Générale de Banque (64 Kbps, 19.2 Kbps et 9.6 Kbps) (cfr figure 2.1 page 48).



LEGENDE :

- 
 -----> NOEUD SL-10
- > TRONC A 9600 BPS
- - - - -> TRONC A 19200 BPS
- > TRONC A 64000 BPS.

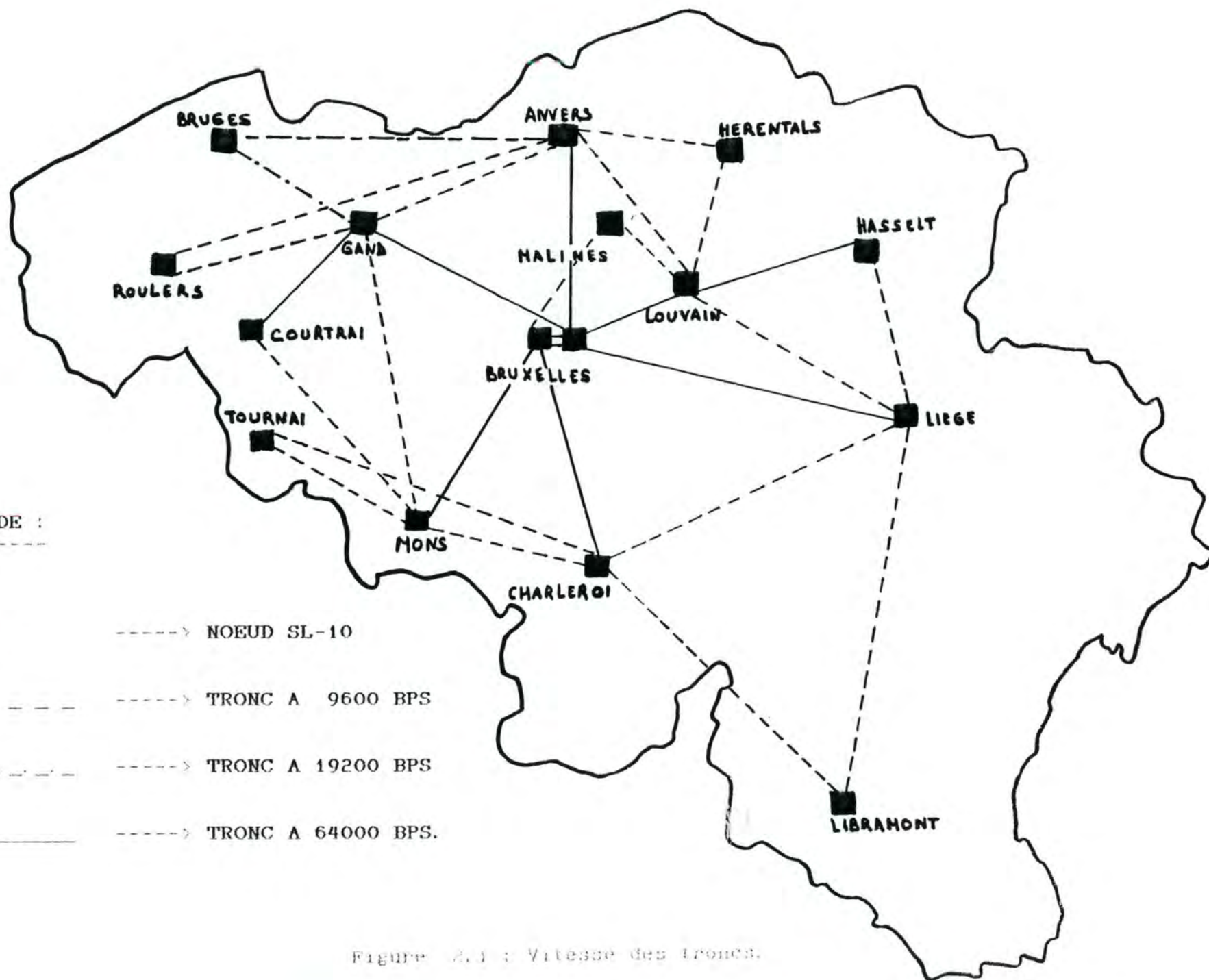


Figure 2.1 : Vitesse des troncs.



### 2.2.3 Algorithme de routage : ROUT.

Chaque tronç possède un poids en fonction de sa vitesse :

64 Kbps	----->	le poids vaut 1
19.2 Kbps	----->	le poids vaut 2
9.6 Kbps	----->	le poids vaut 5

Un algorithme de routage examine tous les chemins possibles pour relier tous les noeuds entre eux. Pour chaque chemin, il calcule la somme des valeurs des troncs le constituant et choisit ensuite celui qui a la plus petite valeur. Cette valeur ne peut excéder 15, sinon la communication ne peut être établie.

Cet algorithme entraîne une quasi non-utilisation de certains troncs (cfr figure 2.2 page 50). On peut donc diviser les troncs en deux catégories : les troncs utilisés couramment par l'algorithme de routage et ceux qui sont uniquement employés en cas de panne d'un tronç quelconque du premier type.

Exemple : Un utilisateur relié au noeud de Charleroi veut parler à une autre personne reliée quant à elle au noeud de Mons. Deux chemins sont possibles (Charleroi-Mons avec un poids égal à 5 et Charleroi-Bruxelles-Mons avec un poids égal à 2 (cfr figure 2.1 page 48)).

L'algorithme de routage choisira toujours le dernier chemin car sa valeur est la plus petite des deux. La communication passera donc toujours par le noeud situé à Bruxelles. Le tronç Charleroi-Mons sera uniquement utilisé si l'un des troncs (Mons-Bruxelles ou Bruxelles-Charleroi) tombe en panne.

### 2.2.4 Type du noeud origine du tronç : NOEUDORI.

Il s'agit du type du noeud origine du tronç concerné par la prise des mesures. Pour le type d'application 'Agences', ce type est celui du noeud reliant ces dernières au réseau. Pour le type d'application 'Front-end', il s'agit du type du noeud reliant celui-ci au réseau.

### 2.2.5 Type du noeud destination du tronç : NOEUDDES.

Il s'agit du type du noeud destination du tronç concerné par la prise des mesures. Pour le type d'application 'Agences', ce type est celui du noeud reliant le front-end dont ces dernières dépendent. Pour le type d'application 'Front-end', il s'agit du numéro de chaque noeud reliant les agences dépendant de celui-ci.

### 2.2.6 Nombre de segments : SEG.

Il s'agit du nombre de segments qui doivent transiter d'un utilisateur vers un autre durant la durée de l'appel. Cette variable sera mesurée au même moment que le temps d'existence d'un appel. En effet, les deux valeurs se retrouvent dans le même enregistrement du fichier de comptabilité des appels 'ACNT. DAT'.





### 2.2.7 Configuration des noeuds : FEN et AGEN.

A partir de notre connaissance acquise à ce stade de l'étude, il nous était possible de présenter des tableaux représentant par noeud, les deux principaux types d'utilisateurs. En effet, nous savions, grâce au fonctionnement du réseau, que les agences communiquaient constamment avec les front-ends et inversement.

* * * * *			* * * * *			* * * * *			* * * * *		
*                      *			*                      *			*                      *			*                      *		
*    Numéro        du    *			*    Nombre        de    *			*    Nombre        *			*    *		
*        noeud        *			*   front-ends (FEN)   *			*   d'agences (AGEN)   *			*    *		
*                      *			*                      *			*                      *			*                      *		
* * * * *			* * * * *			* * * * *			* * * * *		
*                      *			*                      *			*                      *			*                      *		
*        001 (80)        *			*            1            *			*        66            *			*                      *		
*        11                *			*            0            *			*        42            *			*                      *		
*        12                *			*            0            *			*        71            *			*                      *		
*        13                *			*            0            *			*        49            *			*                      *		
*        23                *			*            0            *			*        21            *			*                      *		
*        24                *			*            0            *			*        90            *			*                      *		
*        30                *			*            0            *			*        36            *			*                      *		
*        31                *			*            0            *			*        36            *			*                      *		
*        32                *			*            0            *			*        25            *			*                      *		
*        34                *			*            0            *			*        35            *			*                      *		
*        81                *			*            0            *			*        157            *			*                      *		
*        82                *			*            1            *			*        156            *			*                      *		
*        83                *			*            1            *			*        102            *			*                      *		
*        84                *			*            1            *			*        46            *			*                      *		
*        85                *			*            1            *			*        117            *			*                      *		
*        86                *			*            1            *			*        74            *			*                      *		
*                      *			*                      *			*                      *			*                      *		
* * * * *			* * * * *			* * * * *			* * * * *		

### 2.2.8 Nombre d'agences dépendant du front-end situé sur le noeud étudié : AF.

Chaque agence dépend d'un et un seul des six front-ends batch. Chaque front-end gère donc un nombre déterminé d'agences. Durant la journée, plus ce nombre est élevé, plus le front-end recevra d'informations en provenance de ses agences. En fin de nuit, plus ce nombre est grand, plus le front-end enverra de messages de mise à jour des bases de données de ses agences. Cette variable vaut zéro si le noeud étudié ne contient pas de front-end.

Le nombre d'agences par front-end, ainsi que le lieu d'implantation (cfr figure 2.3 page 53) ont été fournis au moyen d'une procédure écrite en DATATRIEVE 'NOMB\_AGENCES' présentée en annexe page 24.



## \* NOMBRE D'AGENCES PAR FRONT-END \*

RESULTAT DU FICHIER DU 3 AU 4 DECEMBRE 1985

80002007

NOMBRE D'AGENCES AU NOEUD 80	:	31
NOMBRE D'AGENCES AU NOEUD 81	:	120
		***
TOTAL	:	151

82002009

NOMBRE D'AGENCES AU NOEUD 32	:	1
NOMBRE D'AGENCES AU NOEUD 80	:	24
NOMBRE D'AGENCES AU NOEUD 81	:	8
NOMBRE D'AGENCES AU NOEUD 82	:	144
NOMBRE D'AGENCES AU NOEUD 83	:	19
		***
TOTAL	:	196

83002007

NOMBRE D'AGENCES AU NOEUD 11	:	42
NOMBRE D'AGENCES AU NOEUD 13	:	48
NOMBRE D'AGENCES AU NOEUD 31	:	36
NOMBRE D'AGENCES AU NOEUD 83	:	83
		***
TOTAL	:	209

84002011

NOMBRE D'AGENCES AU NOEUD 12	:	71
NOMBRE D'AGENCES AU NOEUD 30	:	36
NOMBRE D'AGENCES AU NOEUD 32	:	24
NOMBRE D'AGENCES AU NOEUD 80	:	5
NOMBRE D'AGENCES AU NOEUD 81	:	4
NOMBRE D'AGENCES AU NOEUD 82	:	12
NOMBRE D'AGENCES AU NOEUD 84	:	46
		***
TOTAL	:	198

85002001

NOMBRE D'AGENCES AU NOEUD 24	:	30
NOMBRE D'AGENCES AU NOEUD 34	:	35
NOMBRE D'AGENCES AU NOEUD 80	:	1
NOMBRE D'AGENCES AU NOEUD 81	:	1
NOMBRE D'AGENCES AU NOEUD 85	:	117
		***
TOTAL	:	184

86002011

NOMBRE D'AGENCES AU NOEUD 13	:	1
NOMBRE D'AGENCES AU NOEUD 23	:	21
NOMBRE D'AGENCES AU NOEUD 24	:	60
NOMBRE D'AGENCES AU NOEUD 80	:	5
NOMBRE D'AGENCES AU NOEUD 81	:	24
NOMBRE D'AGENCES AU NOEUD 86	:	74
		***
TOTAL	:	185

NOMBRE TOTAL D'AGENCES : 1123

LEGENDE :

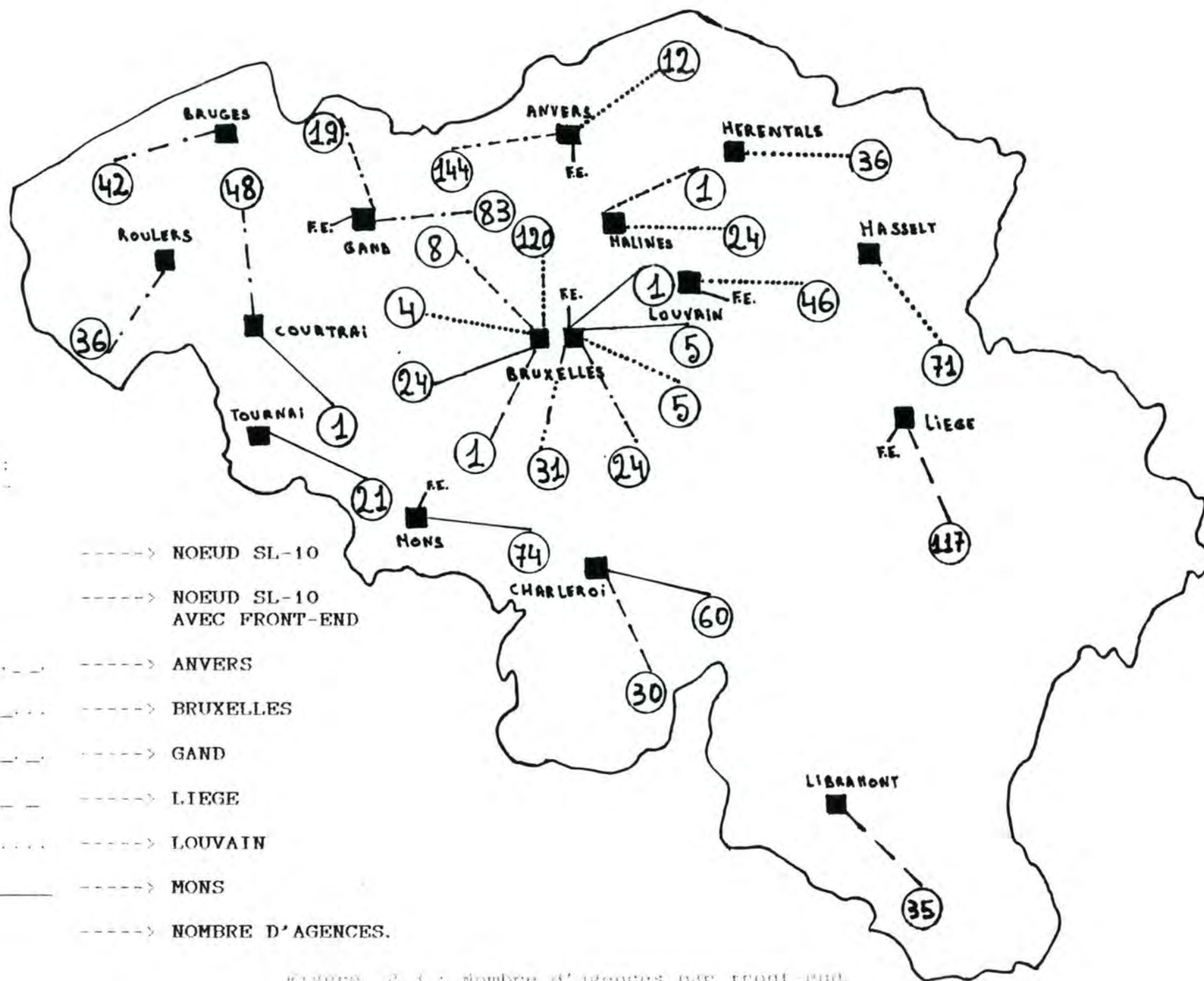
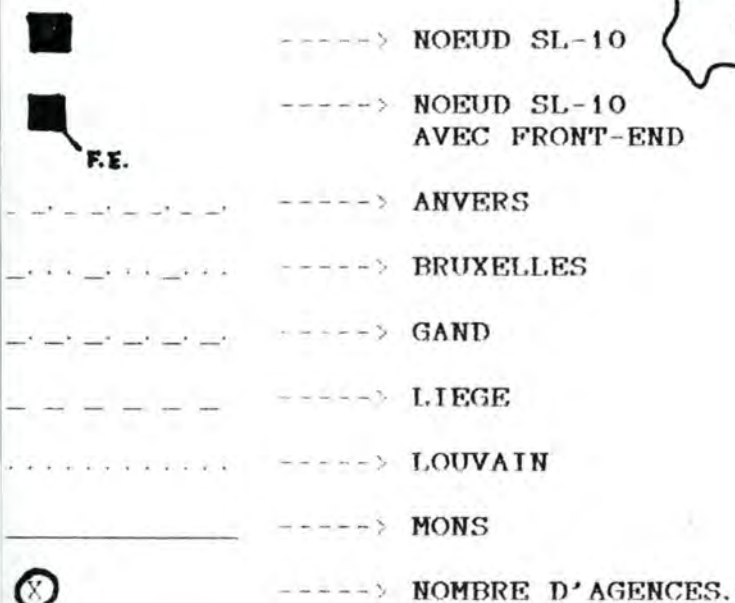


Figure 2.3 : Nombre d'agences par front-end.

2.2.9 Nombre d'agences reliées au noeud étudié et dépendant du front-end situé sur ce noeud : **AFN**.

Cette variable vaut zéro lorsque le noeud étudié ne comporte pas de front-end.

2.2.10 Nombre d'agences reliées au noeud étudié et dépendant d'un front-end non situé sur ce noeud : **AAF**

Cette variable est toujours non nulle lorsque le noeud étudié ne comporte pas de front-end.

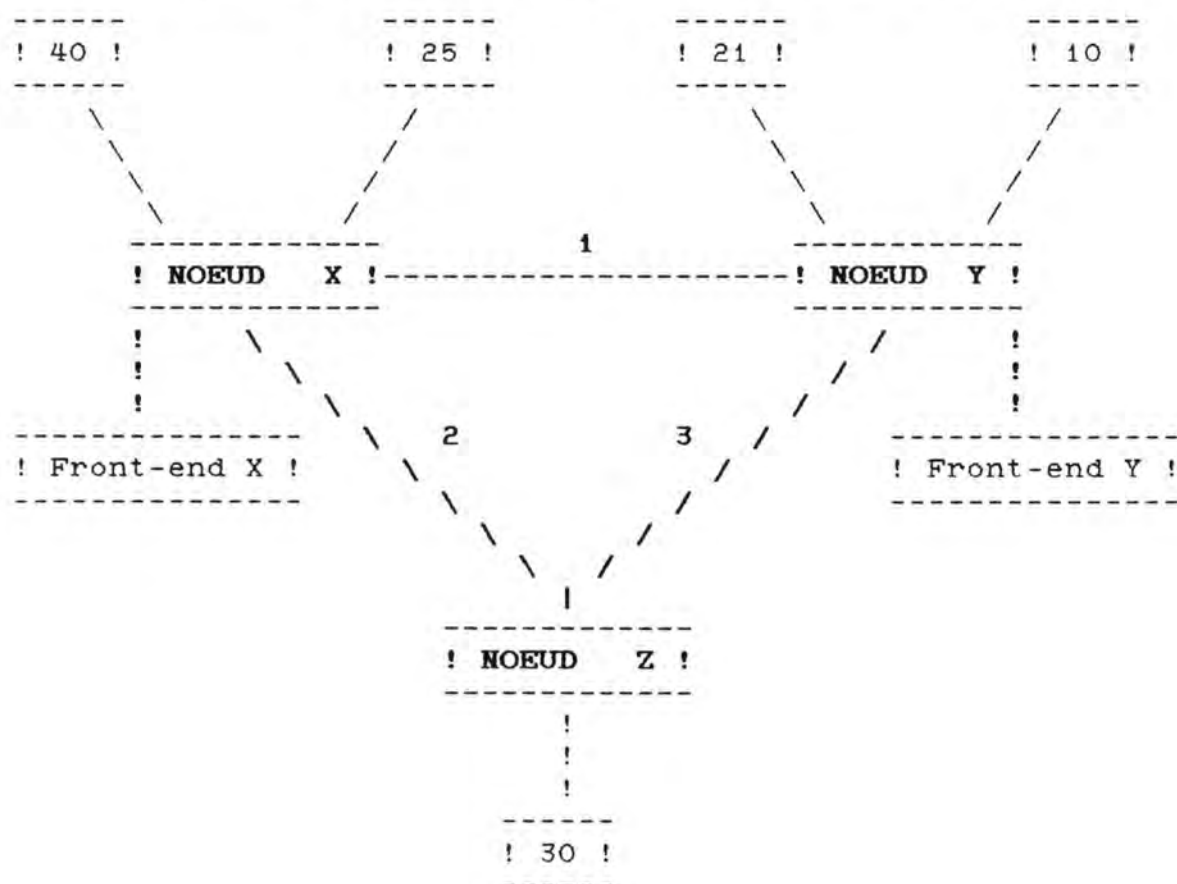
2.2.11 Nombre d'agences reliées à un autre noeud et dépendant du front-end situé sur le noeud étudié : **AFPN**.

Cette variable vaut zéro si le noeud étudié ne comporte pas de front-end.

Pour justifier l'utilisation de ces trois variables, il est nécessaire d'employer un graphique explicatif.

Soit

- 3 noeuds X, Y et Z
- 3 troncs 1, 2 et 3
- 1 front-end sur le noeud X (FEX)
- 1 front-end sur le noeud Y (FEY)
- 40 agences du noeud X dépendant du front-end FEY
- 25 agences du noeud X dépendant du front-end FEX
- 21 agences du noeud Y dépendant du front-end FEY
- 10 agences du noeud Y dépendant du front-end FEX
- 30 agences du noeud Z dépendant du front-end FEX.





Si on considère le noeud X, on peut décomposer les agences en trois groupes :

- Nombre d'agences reliées au noeud X et dépendant de FEX.  
Plus ce nombre est grand, plus le pourcentage de segments locaux au noeud X sera élevé.

Exemple : 25 agences.

- Nombre d'agences reliées à un autre noeud que X et dépendant de FEX.

Exemple : 10 + 30 agences.

- Nombre d'agences reliées au noeud X et dépendant de FEY.  
Plus ce nombre est grand, plus le pourcentage de segments locaux au noeud X sera faible.

Exemple : 40 agences.

Voici les résultats pour le réseau de la Générale de Banque :

Noeud	Nombre de front-ends	Variable AFN	Variable AFPN	Variable AAF
001	1	31	120	35
11	0	0	0	42
12	0	0	0	71
13	0	0	0	49
23	0	0	0	21
24	0	0	0	90
30	0	0	0	36
31	0	0	0	36
32	0	0	0	25
34	0	0	0	35
81	0	0	0	157
82	1	144	52	12
83	1	83	126	19
84	1	46	152	0
85	1	117	67	0
86	1	74	111	0

#### 2.2.12 Présence d'agences "mère" et d'agences "fille" : AGENMF.

Sur certains noeuds de la Générale de Banque, se trouvent des agences qui communiquent fréquemment entre elles, sans passer par les services d'un front-end.

Il en existe actuellement sur trois noeuds : BRUXELLES 1 (80)  
BRUXELLES 2 (81)  
CHARLEROI (24).

Après avoir décrit les variables explicatives se rapportant à la présentation globale des noeuds et au fonctionnement du réseau, nous pouvons descendre à un niveau plus fin de décomposition : celui des processeurs et en particulier des processeurs de lignes.

### 2.3 VARIABLES EXPLICATIVES DES MESURES EFFECTUEES SUR LES PROCESSEURS.

#### 2.3.1 Configuration des processeurs de lignes.

Chaque processeur de lignes est configuré de façon déterminée. Il permet de relier un certain nombre de types d'utilisateurs au réseau au moyen de lignes asynchrones et synchrones. Nous pouvons présenter sous forme de tableau la configuration existante pour les trente différents processeurs de lignes de tout le réseau de la Générale de Banque.

#### Noeud 001 (80).

```

*****
*   Processeur 9   *   Processeur 10   *   Processeur 11   *   Processeur 12   *
*****
*                 *                 *                 *                 *
* 30 agences      * 35 agences      * 1 agence        *                 *
* 5 bureautique   *                 * 30 bureautique  *                 *
* 2 ITI           * 8 ITI           *                 *                 *
* 1 Front-end     * 1 F-E test      * 1 F-E test     *                 *
* 1 Hqr80         *                 *                 *                 *
* 4 intercer      *                 * 2 intercer     *                 *
* 1 portefeuille *                 *                 *                 *
* 4 salles cours  *                 *                 *                 *
* 12 test Jos 2   *                 * 1 test Jos 2   *                 *
*                 *                 *                 *                 *
*****

```

#### Noeud 11.

```

*****
*   Processeur 9   *   Processeur 10   *   Processeur 11   *   Processeur 12   *
*****
*                 *                 *                 *                 *
* 42 agences      *                 *                 *                 *
* 2 bureautique   *                 *                 *                 *
* 2 ITI           *                 *                 *                 *
* 1 salles cours  *                 *                 *                 *
*                 *                 *                 *                 *
*****

```



Noeud 12.

```

*****
*   Processeur 9   *   Processeur 10   *   Processeur 11   *   Processeur 12   *
*****
*
* 43 agences      * 28 agences      *
* 1 bureautique  *
* 2 ITI          *
*                * 1 salle cours *
*
*****

```

Noeud 13.

```

*****
*   Processeur 9   *   Processeur 10   *   Processeur 11   *   Processeur 12   *
*****
*
* 49 agences      *
* 1 bureautique  *
* 2 ITI          *
* 1 salle cours  *
*
*****

```

Noeud 23.

```

*****
*   Processeur 9   *   Processeur 10   *   Processeur 11   *   Processeur 12   *
*****
*
* 21 agences      *
* 1 bureautique  *
* 2 ITI          *
*
*****

```

Noeud 24.

```

*****
*   Processeur 9   *   Processeur 10   *   Processeur 11   *   Processeur 12   *
*****
*
* 43 agences      * 47 agences      *
* 1 bureautique  * 1 bureautique  *
* 2 ITI          *
*                * 2 salle cours  *
*
*****

```



Noeud 30.

```

*****
*   Processeur 9   *   Processeur 10   *   Processeur 11   *   Processeur 12   *
*****
*                 *                 *                 *                 *
* 36 agences      *                 *                 *                 *
* 1 bureautique   *                 *                 *                 *
* 2 ITI           *                 *                 *                 *
*                 *                 *                 *                 *
*****

```

Noeud 31.

```

*****
*   Processeur 9   *   Processeur 10   *   Processeur 11   *   Processeur 12   *
*****
*                 *                 *                 *                 *
* 36 agences      *                 *                 *                 *
* 2 ITI           *                 *                 *                 *
*                 *                 *                 *                 *
*****

```

Noeud 32.

```

*****
*   Processeur 9   *   Processeur 10   *   Processeur 11   *   Processeur 12   *
*****
*                 *                 *                 *                 *
* 25 agences      *                 *                 *                 *
* 1 bureautique   *                 *                 *                 *
* 2 ITI           *                 *                 *                 *
*                 *                 *                 *                 *
*****

```

Noeud 34.

```

*****
*   Processeur 9   *   Processeur 10   *   Processeur 11   *   Processeur 12   *
*****
*                 *                 *                 *                 *
* 35 agences      *                 *                 *                 *
* 1 bureautique   *                 *                 *                 *
* 2 ITI           *                 *                 *                 *
* 1 salle cours   *                 *                 *                 *
*                 *                 *                 *                 *
*****

```

Noeud 81.

```

*****
*   Processeur 9   *   Processeur 10   *   Processeur 11   *   Processeur 12   *
*****
*
* 38 agences      * 55 agences      * 52 agences      * 12 agences      *
*                 *                 * 7 bureautique   *                 *
* 3 Ccnc          *                 *                 *                 *
*                 *                 *                 * 1 F-E test      *
*                 *                 *                 * 2 Gline         *
* 1 Hqc81         *                 *                 *                 *
*                 *                 * 1 intercer      * 1 intercer      *
* 8 ITI           *                 *                 *                 *
*                 *                 *                 * 1 salle cours   *
*                 * 2 test Jos 2   *                 * 2 test Jos 2    *
*                 *                 *                 *                 *
*****

```

Noeud 82.

```

*****
*   Processeur 9   *   Processeur 10   *   Processeur 11   *   Processeur 12   *
*****
*
* 39 agences      * 60 agences      * 57 agences      *                 *
* 1 bureautique   *                 * 1 bureautique   *                 *
*                 * 1 Front-end     *                 *                 *
* 4 ITI           *                 *                 *                 *
* 1 salle cours   *                 *                 *                 *
*                 *                 *                 *                 *
*****

```

Noeud 83.

```

*****
*   Processeur 9   *   Processeur 10   *   Processeur 11   *   Processeur 12   *
*****
*
* 42 agences      * 60 agences      *                 *                 *
* 2 bureautique   *                 *                 *                 *
* 1 Front-end     *                 *                 *                 *
* 5 ITI           *                 *                 *                 *
* 2 salle cours   *                 *                 *                 *
*                 *                 *                 *                 *
*****

```



Noeud 84.

```

*****
*   Processeur 9   *   Processeur 10   *   Processeur 11   *   Processeur 12   *
*****
*
* 22 agences      * 24 agences      *
* 1 bureautique   * 1 bureautique   *
* 1 Front-end     *
* 4 ITI           *
* 1 salle cours   *
*
*****

```

Noeud 85.

```

*****
*   Processeur 9   *   Processeur 10   *   Processeur 11   *   Processeur 12   *
*****
*
* 53 agences      * 60 agences      * 4 agences      *
*                 *                 * 3 bureautique  *
* 1 Front-end     *
* 4 ITI           *
* 1 salle cours   * 1 salle cours   *
*
*****

```

Noeud 86.

```

*****
*   Processeur 9   *   Processeur 10   *   Processeur 11   *   Processeur 12   *
*****
*
* 36 agences      * 38 agences      *
* 1 bureautique   * 1 bureautique   *
* 1 Front-end     *
* 1 intercer      *
* 4 ITI           *
* 1 salle cours   *
*
*****

```

Remarque :

\* modification à partir du 27 novembre 1985 :

Au noeud 001, le Front-end de test passe sur le processeur 9, le Front-end passe sur le processeur 10 et l'HQR80 passe sur le processeur 11.

\* modification à partir du 10 janvier 1986 :

Aux noeuds 83, 84 et 86, les Front-ends passent sur le processeur 10.



Expliquons brièvement les différents types d'utilisateurs rencontrés dans ces tableaux :

\* **Agences (AL).**  
-----

Voir explication au chapitre 1 page 13.

\* **Front-ends (F).**  
-----

Voir explication au chapitre 1 page 13.

\* **Call01 (I).**  
-----

Voir explication au chapitre 1 page 16.

\* **CCNC (C).**  
-----

Ce sont deux machines (PDP11/44A et PDP11/44B) qui communiquent avec l'HQR80 et l'HQC81. Le PDP11/44A gère une base de données. Sur le PDP11/44B, on développe un opérateur automatique.

\* **PORTEFEUILLE (P).**  
-----

Machine destinée aux opérations portefeuille étranger. Pendant la journée, elle ne dispose pas de place suffisante pour stocker les logiciels relatifs aux protocoles de réseau. En raison de cette contrainte, les différentes transactions sont gérées localement.

En fin d'après-midi, vers 16 - 17 heures, lorsque les fonctions bancaires se clôturent, la place mémoire se libère et peut être ainsi utilisée pour installer les protocoles d'accès. La machine, en tant qu'utilisateur, se connecte alors au réseau via le noeud de Bruxelles et envoie toutes les informations collectées durant la journée.

\* **GLINE (G).**  
-----

Ce sont deux machines connectées au noeud 81 de Bruxelles, offrant des services spéciaux à certains clients de la Générale de Banque.

Lorsqu'un de ces clients désire connaître sa situation personnelle, il utilise le réseau public téléphonique pour accéder à la base de données contenant les informations nécessaires.

\* **HQR80 (HR).**  
-----

Machine connectée au noeud 80 de Bruxelles, dont les fonctions principales sont les suivantes :

- **Transfert de fichiers et envoi de nouveaux programmes vers les agences.**

La politique de distribution des différents logiciels est totalement aléatoire. En effet, si plusieurs fichiers FICHIER1, FICHIER2 et FICHIER3 doivent être expédiés vers les agences AGENCE1, AGENCE2 et AGENCE3, l'HQR80 envoie dans un premier temps le fichier FICHIER1. Si l'AGENCE1 signale que la réception du FICHIER1 est totalement terminée, cela n'implique pas nécessairement que le FICHIER2 va lui être directement envoyé.

- **Exécution de commandes à distance durant la nuit pour connaître à titre d'exemple, l'état de la base de données locale se trouvant dans une agence particulière. Il est à signaler que cette deuxième fonction est réalisée beaucoup plus rarement que la première.**

\* **HQC81 (HC).**  
-----

Machine connectée au noeud 81 de Bruxelles, réalisant principalement les deux fonctions suivantes :

- **Contrôle à distance durant la journée.**

Plus ou moins 20.000 à 30.000 commandes sont réalisées par jour. De 11 à 12 heures, l'HQC81 demande par exemple l'heure aux différentes agences afin de déceler un problème quelconque dans la chaîne élémentaire de transmission ainsi requise.

- **Contrôle de la mise à jour des bases de données.**

Pendant la nuit ou très tôt le matin (de toute façon avant 6 heures), les opérateurs de l'HQC81 interrogent les agences reliées au réseau afin de savoir si la mise à jour des bases de données locales s'est déroulée correctement.



# \* FRONT-END DE TEST (T).

---

Il s'agit de trois machines connectées aux noeuds de Bruxelles et utilisées pour effectuer certaines procédures de test. En effet, les logiciels destinés à être implémentés dans les six centres électroniques sont d'abord testés sur ces trois ordinateurs. Lorsque les tests s'avèrent satisfaisants, les logiciels contrôlés peuvent alors devenir effectivement opérationnels.

# \* TEST RUE JOSEPH 2 (J).

---

Machines connectées aux noeuds de Bruxelles, utilisées pour réaliser des procédures de test. En effet, avant d'implémenter directement différentes applications sur le réseau, celles-ci sont d'abord testées par des analystes-programmeurs dépendant du centre de la rue Joseph 2 à Bruxelles.

# \* SALLES DE COURS (S).

---

Machines situées dans toute la Belgique et qui permettent aux utilisateurs futurs du réseau (par exemple : les employés d'agences) de se familiariser au matériel informatique de la banque. Durant certaines heures de cours, un équipement d'essai est ainsi mis à la disposition de ces personnes, ce qui a pour effet une insertion ultérieure beaucoup plus aisée, rapide et opérationnelle du personnel.

# \* INTERCER (IN).

---

Le but de ce type d'application est de faire supporter par le réseau maillé de la Générale de Banque la communication entre les six centres électroniques. Actuellement, on utilise uniquement la liaison entre le front-end de Bruxelles et celui de Mons. Le reste du trafic utilise toujours les services fournis par un réseau parallèle reliant les six centres entre eux.

# \* BUREAUTIQUE (B).

---

Ce sont des machines recouvrant un ensemble d'outils de gestion pour l'information de bureau. La principale fonction utilisatrice du réseau est le transfert de fichiers avec une machine se trouvant sur le noeud 80 (archivage).



2.3.2 Nombre d'agences dépendant du front-end situé sur le processeur étudié et se trouvant sur celui-ci : A.

Plus ce nombre est grand, plus il y aura de trafic à l'intérieur du processeur de lignes étudié.

En effet, durant la journée, le processeur devra s'occuper **en même temps** :

- de l'envoi des données à partir des agences de ce type vers le front-end
- et de la réception des informations transmises par toutes les agences dépendant du front-end.

Durant la nuit, ce sera la même chose, mais la circulation des mises à jour se fera dans le sens opposé. Il existe dès lors une utilisation double du processeur (par rapport à un processeur sans front-end) durant toute la journée et toute la nuit pour ce type d'agence.

Cette variable vaut zéro si le processeur en question ne supporte pas de front-end.

* * * * *			* * * * *			* * * * *			* * * * *		
* Numéro du			* Numéro du			* Nombre d'agences			* A		
* noeud			* processeur			* sur le processeur			* AL		
* * * * *			* * * * *			* * * * *			* * * * *		
* 001 (80) *			* 09 *			* 30 *			* 0 *		
* 001 (80) *			* 10 *			* 35 *			* 30 *		
* 001 (80) *			* 11 *			* 1 *			* 0 *		
* 11 *			* 09 *			* 42 *			* 0 *		
* 12 *			* 09 *			* 43 *			* 0 *		
* 12 *			* 10 *			* 28 *			* 0 *		
* 13 *			* 09 *			* 49 *			* 0 *		
* 23 *			* 09 *			* 21 *			* 0 *		
* 24 *			* 09 *			* 43 *			* 0 *		
* 24 *			* 10 *			* 47 *			* 0 *		
* 30 *			* 09 *			* 36 *			* 0 *		
* 31 *			* 09 *			* 36 *			* 0 *		
* 32 *			* 09 *			* 25 *			* 0 *		
* 34 *			* 09 *			* 35 *			* 0 *		
* 81 *			* 09 *			* 38 *			* 0 *		
* 81 *			* 10 *			* 55 *			* 0 *		
* 81 *			* 11 *			* 52 *			* 0 *		
* 81 *			* 12 *			* 12 *			* 0 *		
* 82 *			* 09 *			* 39 *			* 0 *		
* 82 *			* 10 *			* 60 *			* 54 *		
* 82 *			* 11 *			* 57 *			* 0 *		
* 83 *			* 09 *			* 42 *			* 0 *		
* 83 *			* 10 *			* 60 *			* 47 *		
* 84 *			* 09 *			* 22 *			* 0 *		
* 84 *			* 10 *			* 24 *			* 24 *		
* 85 *			* 09 *			* 53 *			* 53 *		
* 85 *			* 10 *			* 60 *			* 0 *		
* 85 *			* 11 *			* 4 *			* 0 *		
* 86 *			* 09 *			* 36 *			* 0 *		
* 86 *			* 10 *			* 38 *			* 38 *		
* * * * *			* * * * *			* * * * *			* * * * *		

2.3.3 Nombre d'agences se trouvant sur le processeur étudié et ne dépendant pas du front-end situé sur celui-ci : AMP.

Cette variable s'obtient en faisant la différence entre le nombre d'agences reliées au processeur étudié (AL) et le contenu de la variable précédente relative au même processeur. Sa valeur est toujours non nulle lorsqu'il n'y a pas de front-end (si F = 0).

Pour ce type d'agence, le processeur ne doit uniquement s'occuper que du transfert des informations dans un sens en même temps (le jour : agences -> front-end et la nuit front-end -> agences).

* * * * *	* * * * *	* * * * *	* * * * *
* Numéro du	* Numéro du	* AL	* A
* noeud	* processeur		
* * * * *	* * * * *	* * * * *	* * * * *
* 001 (80)	* 09	* 30	*
* 001 (80)	* 10	* 5	*
* 001 (80)	* 11	* 1	*
* 11	* 09	* 42	*
* 12	* 09	* 43	*
* 12	* 10	* 28	*
* 13	* 09	* 49	*
* 23	* 09	* 21	*
* 24	* 09	* 43	*
* 24	* 10	* 47	*
* 30	* 09	* 36	*
* 31	* 09	* 36	*
* 32	* 09	* 25	*
* 34	* 09	* 35	*
* 81	* 09	* 38	*
* 81	* 10	* 55	*
* 81	* 11	* 52	*
* 81	* 12	* 12	*
* 82	* 09	* 39	*
* 82	* 10	* 6	*
* 82	* 11	* 57	*
* 83	* 09	* 42	*
* 83	* 10	* 13	*
* 84	* 09	* 22	*
* 84	* 10	* 0	*
* 85	* 09	* 0	*
* 85	* 10	* 60	*
* 85	* 11	* 4	*
* 86	* 09	* 36	*
* 86	* 10	* 0	*
* * * * *	* * * * *	* * * * *	* * * * *

### 2.3.4 Buffer disponible : BU.

Chaque processeur de lignes doit disposer d'un certain nombre de bytes en mémoire (appelé "buffer") afin de permettre l'établissement de circuits virtuels aux utilisateurs qu'il relie au réseau. La taille de ce "buffer" varie de processeur à processeur en fonction du nombre et des types d'utilisateurs.

A partir de la configuration d'un processeur, il est possible de calculer la dimension exacte de cette zone en mémoire.

Plus il y a d'utilisateurs, plus le "buffer" est petit et donc plus le nombre de circuits virtuels permis à un instant donné est faible.

Les différents chiffres exprimant la taille des "buffers" et par conséquent le nombre de circuits permis ont été fournis par une procédure écrite en MACRO.

* * * * *					* * * * *					* * * * *					* * * * *				
* * * * *					* * * * *					* * * * *					* * * * *				
* Numéro du					* Numéro du					* Taille du					* Nombre de circuits				
* noeud					* processeur					* "buffer"					* virtuels permis				
* * * * *					* * * * *					* * * * *					* * * * *				
* * * * *					* * * * *					* * * * *					* * * * *				
* * * * *					* * * * *					* * * * *					* * * * *				
* * * * *					* * * * *					* * * * *					* * * * *				
* * * * *					* * * * *					* * * * *					* * * * *				
* 001 (80) *					* 09 *					* 9087 *					* 284 *				
* 001 (80) *					* 10 *					* 8497 *					* 266 *				
* 001 (80) *					* 11 *					* 37673 *					* 1177 *				
* 11 *					* 09 *					* 12527 *					* 391 *				
* 12 *					* 09 *					* 10925 *					* 341 *				
* 12 *					* 10 *					* 38374 *					* 1199 *				
* 13 *					* 09 *					* 11033 *					* 345 *				
* 23 *					* 09 *					* 18005 *					* 563 *				
* 24 *					* 09 *					* 10925 *					* 341 *				
* 24 *					* 10 *					* 35137 *					* 1098 *				
* 30 *					* 09 *					* 12863 *					* 402 *				
* 31 *					* 09 *					* 15017 *					* 469 *				
* 32 *					* 09 *					* 15602 *					* 488 *				
* 34 *					* 09 *					* 14243 *					* 445 *				
* 81 *					* 09 *					* 8659 *					* 271 *				
* 81 *					* 10 *					* 32668 *					* 1021 *				
* 81 *					* 11 *					* 32448 *					* 1014 *				
* 81 *					* 12 *					* 43170 *					* 1349 *				
* 82 *					* 09 *					* 10564 *					* 330 *				
* 82 *					* 10 *					* 32648 *					* 1020 *				
* 82 *					* 11 *					* 33643 *					* 1051 *				
* 83 *					* 09 *					* 8821 *					* 276 *				
* 83 *					* 10 *					* 32448 *					* 1014 *				
* 84 *					* 09 *					* 15955 *					* 499 *				
* 84 *					* 10 *					* 41860 *					* 1308 *				
* 85 *					* 09 *					* 9286 *					* 290 *				
* 85 *					* 10 *					* 32448 *					* 1014 *				
* 85 *					* 11 *					* 45346 *					* 1417 *				
* 86 *					* 09 *					* 11037 *					* 345 *				
* 86 *					* 10 *					* 37926 *					* 1185 *				
* * * * *					* * * * *					* * * * *					* * * * *				
* * * * *					* * * * *					* * * * *					* * * * *				



Après avoir présenté les différentes mesures à effectuer, nous avons construit un tableau reprenant d'une part les variables à expliquer et d'autre part les variables qui selon notre opinion, influençaient ces dernières.

Variables à expliquer	Variables explicatives
Pourcentage d'utilisation des processeurs de lignes (TAUXPL)	-Type d'application (AP) -Buffer disponible (BU) -Nombre d'agences sur le processeur (AL) -Nombre d'agences reliées au front-end (AF) -Nombre de bureaux sur le processeur (B) -Nombre de front-ends sur le processeur (F) -Nombre de ccnc sur le processeur (C) -Nombre de front-ends de test sur le processeur (T) -Nombre de rj2test sur le processeur (J) -Nombre de call01 (ITI) sur le processeur (I) -Nombre d' hqr80 sur le processeur (HR) -Nombre d' hqc81 sur le processeur (HC) -Nombre de salles de cours sur le processeur (S) -Nombre d' intercer sur le processeur (IN) -Nombre d'agences sur le même processeur que le front-end dont elles dépendent (A) -Nombre d'agences se trouvant sur le processeur étudié et ne dépendant pas du front-end situé sur celui-ci (AMP) -Nombre de portefeuille sur le processeur (P) -Nombre de gline sur le processeur (G)







### CHAPITRE 3

#### DETERMINATION GRAPHIQUE

#### DES PERIODES D'OBSERVATION.

-----

#### Résumé :

L'objectif de ce chapitre est de se faire une idée de la répartition horaire de la charge (évaluée en nombre de segments) sur l'ensemble du réseau et selon différents types d'application que nous avons déterminés avec l'aide d'une personne ayant une bonne connaissance de son fonctionnement.

Un tel chapitre s'est avéré nécessaire car nous n'avions aucune idée quant à la manière dont le réseau se comportait, et il nous a permis de dégager trois types d'application, considérés comme les principaux consommateurs.

A la fin de ce chapitre, nous sommes donc en mesure de dégager des tranches horaires de charge maximale pour chaque type d'application retenu. De plus, selon l'hypothèse simplificatrice que, à l'intérieur de chaque tranche horaire, toute la charge n'est imputable qu'à un type d'application, nous avons pu étudier les variables du chapitre précédent aux heures les plus chargées, de manière à obtenir un comportement limite du réseau.

-----

### 3.1 OBJECTIF DU CHAPITRE.

N'ayant, à l'époque, aucune idée du comportement général du réseau, nous avons décidé de consacrer quelque temps à cette étude. Dans ce but, nous avons d'abord dégagé les différents types d'application utilisateurs du réseau (Voir chapitre 2, page 61) :

- a) Les Agences
- b) Les Front-ends
- c) Les 'Call01'
- d) Ccnc
- e) Le Portefeuille
- f) Gline
- g) Hqr80 (Headquarter sur le noeud 80)
- h) Hqc81 (Headquarter sur le noeud 81)
- i) Les Front-ends de test
- j) Les machines de test de la rue Joseph 2
- k) Les salles de cours
- l) Les intercer
- m) La bureautique

Les questions auxquelles nous avons essayé de répondre sont les suivantes : - **quelle est la charge totale du réseau ?**  
- **étant donné un type d'application, quelle est sa contribution à cette charge ?**

Pour ce faire, nous avons procédé de la manière suivante :

#### a) Nombre de segments :

- par tranche horaire d'une heure, nous avons déterminé par type d'application, la quantité de segments qui transitaient sur le réseau. Les graphiques obtenus nous ont permis de déterminer l'heure de charge la plus élevée par type d'application.

- nous avons procédé de la même manière pour le comportement global du réseau. Les graphiques obtenus nous ont permis de déterminer l'heure de charge la plus élevée pour l'ensemble du réseau.

#### b) Nombre de 'calls' :

Nous avons procédé de la même manière pour le nombre de 'calls', mais cette information n'a pas pu être directement utilisée, parce que, à ce stade-ci, nous ne connaissions pas le nombre de 'calls' refusés, par exemple.

Nous avons effectué ces mesures durant sept journées, qui sont les suivantes :

- 15/16 octobre 1985 (mardi à mercredi)
- 16/17 octobre 1985 (mercredi à jeudi)
- 21/22 octobre 1985 (lundi à mardi)
- 22/23 octobre 1985 (mardi à mercredi)
- 26/27 octobre 1985 (samedi à dimanche)
- 30/31 octobre 1985 (mercredi à jeudi)
- 31 octobre / 1 novembre 1985 (jeudi à vendredi)

Nous n'insérerons dans ce chapitre que les résultats de la première journée; les résultats concernant les autres journées se trouvent en annexe page 37 et suivantes.



### 3.2 METHODOLOGIE SUIVIE.

#### 3.2.1 Comportement horaire par type d'application.

Il s'agit d'une méthodologie en quatre points :

point a) détermination du comportement horaire du réseau : pour cela, nous avons utilisé des fichiers ('SEGREC.DAT = segments received' et 'SEGSEN.DAT = segments sent', 'CALLNET.DAT = calls network') que nous recevons journallement du DCC (Data Collection Center).

point b) détermination des 'DNA' des différents types d'application.

point c) écriture de procédures en MACRO, afin d'extraire du fichier de comptabilité des appels 'ACNT.DAT', les enregistrements correspondant aux différents types d'application.

point d) écriture de procédures DATATRIEVE permettant d'exploiter les différents fichiers relatifs à chaque type d'application et obtenus au point précédent.

#### Point a.

Les fichiers reçus du DCC regroupent les informations concernant l'ensemble des utilisateurs, mis à part le type d'application 'CALLO1'. Le fichier 'CALLO1.DAT' concernant ce type d'application nous était également fourni par le DCC. Nous devons donc en tenir compte dans nos calculs ultérieurs.

Les fichiers reçus sont les suivants :

Journée du 15/16 octobre 1985 :

* Heure *	* Segments reçus *	* Segments envoyés *	* Total *
* 11 - 12 *	* 50.023 *	* 23.672 *	* 73.695 *
* 12 - 13 *	* 0 *	* 0 *	* 0 *
* 13 - 14 *	* 0 *	* 0 *	* 0 *
* 14 - 15 *	* 44.210 *	* 144.926 *	* 189.136 *
* 15 - 16 *	* 92.265 *	* 314.125 *	* 406.390 *
* 16 - 17 *	* 62.056 *	* 219.314 *	* 281.370 *
* 17 - 18 *	* 58.310 *	* 157.481 *	* 215.791 *
* 18 - 19 *	* 76.409 *	* 200.694 *	* 277.103 *
* 19 - 20 *	* 50.801 *	* 110.395 *	* 161.196 *
* 20 - 21 *	* 143.930 *	* 352.638 *	* 496.568 *
* 21 - 22 *	* 98.149 *	* 285.943 *	* 384.092 *
* 22 - 23 *	* 70.147 *	* 196.610 *	* 266.757 *
* 23 - 24 *	* 74.263 *	* 275.409 *	* 349.672 *
-----			
* 0 - 1 *	* 80.271 *	* 438.169 *	* 518.440 *
* 1 - 2 *	* 135.634 *	* 857.840 *	* 993.474 *
* 2 - 3 *	* 90.633 *	* 547.754 *	* 638.387 *
* 3 - 4 *	* 29.347 *	* 122.067 *	* 151.414 *
* 4 - 5 *	* 41.965 *	* 34.095 *	* 76.060 *
* 5 - 6 *	* 38.675 *	* 57.615 *	* 96.290 *
* 6 - 7 *	* 33.730 *	* 204.325 *	* 238.055 *
* 7 - 8 *	* 33.353 *	* 216.879 *	* 250.232 *
* 8 - 9 *	* 70.473 *	* 156.123 *	* 226.596 *
* 9 - 10 *	* 79.052 *	* 234.911 *	* 313.963 *
* 10 - 11 *	* 83.266 *	* 238.456 *	* 321.722 *
* 11 - 12 *	* 90.934 *	* 294.516 *	* 385.450 *
* 12 - 13 *	* 90.368 *	* 245.975 *	* 336.343 *
* 13 - 14 *	* 54.822 *	* 114.252 *	* 169.074 *
* 14 - 15 *	* 65.192 *	* 194.708 *	* 259.900 *
* 15 - 16 *	* 63.917 *	* 194.336 *	* 258.253 *
=====			
* Total *	* 1.902.195 *	* 6.433.228 *	* 8.335.423 *
*****			

```

*****
*   Heure   *   Nombre de 'calls'   *
*****
*  6 - 7    *           2            *
*  7 - 8    *           0            *
*  8 - 9    *           0            *
*  9 - 10   *           0            *
* 10 - 11   *           0            *
* 11 - 12   *           3            *
* 12 - 13   *           0            *
* 13 - 14   *           0            *
* 14 - 15   *          3.795          *
* 15 - 16   *          7.023          *
* 16 - 17   *          4.760          *
* 17 - 18   *          5.224          *
* 18 - 19   *          4.285          *
* 19 - 20   *          6.772          *
* 20 - 21   *          4.041          *
* 21 - 22   *          7.482          *
* 22 - 23   *          6.118          *
* 23 - 24   *          2.788          *
*-----*
*  0 - 1    *          5.176          *
*  1 - 2    *          3.479          *
*  2 - 3    *          5.146          *
*  3 - 4    *          3.228          *
*  4 - 5    *          5.314          *
*  5 - 6    *          6.924          *
*  6 - 7    *          4.617          *
*  7 - 8    *          3.931          *
*  8 - 9    *          7.253          *
*  9 - 10   *          8.386          *
* 10 - 11   *          9.821          *
* 11 - 12   *          8.538          *
* 12 - 13   *          7.405          *
* 13 - 14   *          5.061          *
* 14 - 15   *          6.597          *
* 15 - 16   *          5.509          *
*=====*
* Total     *         148.678          *
*****

```

Les fichiers concernant les autres jours se trouvent en annexe page 25.



Point b.

Afin de dégager les différents types d'application, nous avons parcouru une liste complète des 'DNA' (Data Network Address), classée par ordre croissant des différents noeuds du réseau. Nous avons alors formé douze ensembles de 'DNA', chaque ensemble correspondant à un type d'application. Les résultats furent les suivants :

a) Front-ends : 6 'DNA'.

- 80002007	82002009	83002007
- 84002011	85002001	86002011

b) Les 'Call01' :

Ils se caractérisent par la présence d'un '0' ou d'un '1' en cinquième position dans le 'DNA'.

c) Ccnc : 3 'DNA'.

- 81002171	81002170
- 81002169	

d) Le Portefeuille : 1 'DNA'.

- 80002024

e) Gline : 2 'DNA'.

- 81002187	81002188
------------	----------

f) Hqr80 : 1 'DNA'.

- 80002003

g) Hqc81 : 1 'DNA'.

- 81002087

h) Front-ends de test : 3 'DNA'.

- 80002027	80002114	81002190
------------	----------	----------

i) Machines de test de la rue Joseph II : 20 'DNA'.

- 80002002	80002009	80002096	80002060
- 80002004	80002099	80002097	80002090
- 80002005	80002066	80002000	81002186
- 80002013	80002074	81002081	80002060
- 80002026	80002075	81002185	81002186

j) Les salles de cours : 18 'DNA'.

- 11002002	34002001	81002172	85002003
- 12002078	80002010	82002001	85002093
- 13002001	80002011	83002002	86002018
- 24002023	80002018	83002006	
- 24002027	80002019	84002009	

k) Les intercer : 9 'DNA'.

K) Les intercer : 9 'DNA'.

- 80002111	80002123	80002014	80002015
- 80002016	80002021	81002189	81002109
- 86002078			

1) Bureautique : 64 'DNA'.

- 11002011	80002128	80002107	81002104
- 11002047	80002129	80002108	81002105
- 12002003	80002130	80002109	81002106
- 13002052	80002131	80002110	81002107
- 23002023	80002132	80002115	81002108
- 24002057	80002134	80002116	82002002
- 24002096	80002135	80002117	82002103
- 30002027	80002136	80002119	83002113
- 32002008	80002137	80002120	83002100
- 34002036	80002100	80002122	84002052
- 80002025	80002101	80002124	84002026
- 80002012	80002102	80002125	85002125
- 80002001	80002103	80002126	85002126
- 80002022	80002104	80002113	85002134
- 80002023	80002105	81002110	86002053
- 80002127	80002106	81002103	86002039

Remarque :

Le type d'application 'AGENCES' n'apparaît pas dans cette liste car les agences sont au nombre de plus ou moins 1.100 (donc plus ou moins 1.100 'DNA'), ce qui était excessif à regrouper.

Les informations sur le nombre de segments et le nombre de 'calls' concernant les agences ont été recueillies par différence entre les fichiers 'SEGREC.DAT', 'SEGSEN.DAT', et 'CALLNET.DAT', et les résultats obtenus pour les douze types précités.

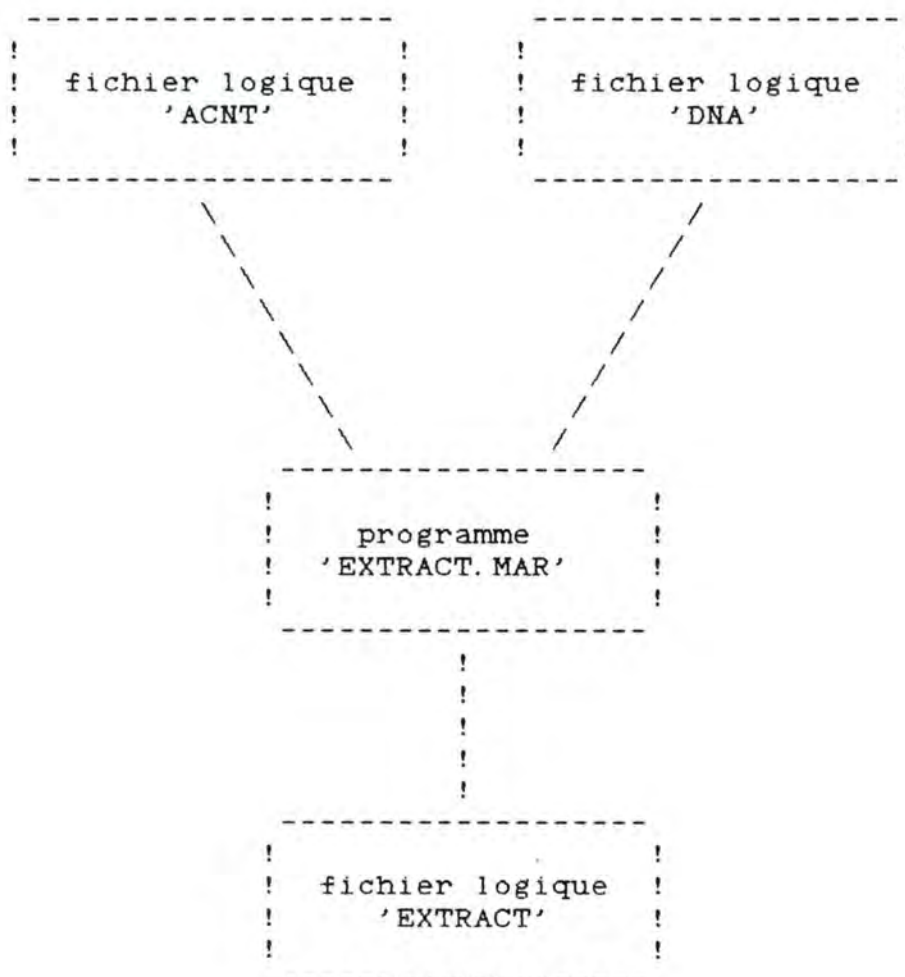
Point c.

- Nous avons écrit en MACRO une procédure d'extraction d'enregistrements du fichier de comptabilité des appels 'ACNT.DAT' (décrit à la page 36) dont les spécifications sont les suivantes :

Procédure 'EXTRACT.MAR' : annexe page 130

Spécification du programme :

Le programme a pour effet d'extraire du fichier de comptabilité des appels 'ACNT.DAT' tous les enregistrements dont le 'DNA' local ou le 'DNA' éloigné est repris dans le fichier logique 'DNA' et de les écrire dans le fichier logique 'EXTRACT'.

Schéma :



### Structure des fichiers :

- *fichiers logiques 'ACNT' et 'EXTRACT' :*

Leur structure est la même que celle du fichier de comptabilité des appels 'ACNT.DAT' décrit à la page 36.

- *fichier logique 'DNA' :*

Il s'agit d'un fichier séquentiel comportant des enregistrements d'une longueur de huit caractères représentant un 'DNA'.

- Ensuite, nous avons écrit en MACRO une procédure par type d'application (dont le nom est 'type d'application.MAR') dont les spécifications sont les suivantes :

exemple pour la procédure 'CCNC.MAR' : annexe page 133

### Spécification du programme :

Le programme a pour effet d'extraire du fichier logique 'ACNT' les enregistrements dont le 'DNA' local est un 'DNA' du type d'application 'CCNC' et dont le 'CALL\_PARAMETERS' vaut 0, 1, 4, 5, 8, 9, 12 ou 13 (Nous ne retenons donc que les enregistrements pour lesquels le 'DNA' local est le 'DNA' appelant du 'call', ce qui signifie que dans la suite, lorsque nous parlerons du type d'application 'CCNC', nous ferons en fait référence à l'ensemble des enregistrements du fichier de comptabilité des appels 'ACNT.DAT' pour lesquels un 'DNA' du type d'application 'CCNC' a établi le 'call'; cette remarque est également valable pour tous les autres types d'application).

Les enregistrements vérifiant ces conditions sont écrits dans le fichier physique 'CCNC.DAT'.



A ce stade-ci, nous disposons donc de douze fichiers ne reprenant que les enregistrements dont le 'DNA' local concerne le type d'application en question et dont le 'CALL\_PARAMETERS' vaut 0, 1, 4, 5, 8, 9, 12 ou 13 (appelants uniquement) :

- FEP. DAT
- CALLO1. DAT
- CCNC. DAT
- PORT. DAT
- GLINE. DAT
- HQR80. DAT
- HQC81. DAT
- FEPTTEST. DAT
- RJ2TEST. DAT
- COURS. DAT
- INTERCER. DAT
- BUR. DAT



Point d.

Sur ces différents fichiers, nous avons exécuté des procédures DATATRIEVE qui nous ont permis de connaître la quantité de segments qui transitent sur le réseau et le nombre de 'calls' relatifs aux différents types d'application.

*Par exemple, pour le fichier 'INTERCER.DAT' :*

- procédure DATATRIEVE **INTERCER** : annexe page 136

L'effet de cette procédure est de fournir par jour et par heure, le nombre de 'calls' relatifs au type d'application 'intercer'.

- procédure DATATRIEVE **INTERCERSEG** : annexe page 137

L'effet de cette procédure est de fournir par jour et par heure, le nombre de segments qui transitent sur le réseau et qui sont relatifs au type d'application 'intercer'.

- procédure DATATRIEVE **INTERCERMEN** : annexe page 138

En exécutant la procédure précédente 'INTERCERSEG', il ne nous était pas possible de tenir compte des enregistrements dus aux applications dont la durée des appels est (ordinairement) très longue (par exemple : 3 heures). Pour ces derniers, nous avons décidé de répartir la charge sur toute la durée du 'call'.

exemple :

Si un 'call' dure trois heures et que 30.000 segments transitent sur le réseau, nous attribuons arbitrairement une charge de 10.000 segments à chaque heure. Cette caractéristique a été observée pour les trois types d'application suivants :

- la bureautique.
- les 'Call01'.
- les intercer.

Les autres types d'application ne sont donc pas concernés par cette remarque.

Les résultats de la journée du 15 au 16 octobre 1985 sont donnés ci-après.

Les résultats des autres journées se trouvent en annexe page 37.

Tableau des différents résultats obtenus :

Journée du 15/16 octobre 1985 (mardi à mercredi) :

1) Type d'application 'CCNC' :

```

*****
*  Heure   *  Nombre   *  Segments   *  Segments   *  Total      *
*          *  'calls'  *  reçus     *  envoyés    *  Segments   *
*****
*  18 - 19 *      3      *    1.124   *    4.709    *    5.833    *
*-----*
*  10 - 11 *      4      *    1.170   *    4.795    *    5.965    *
*=====*
*  Total   *      7      *    2.294   *    9.504    *   11.798    *
*****

```

2) Type d'application 'Front-ends' :

```

*****
*  Heure   *  Nombre   *  Segments   *  Segments   *  Total      *
*          *  'calls'  *  reçus     *  envoyés    *  Segments   *
*****
*  14 - 15 *    640    *    5.422   *   17.929    *   23.351    *
*  15 - 16 *    953    *    8.239   *   30.788    *   39.027    *
*  16 - 17 *    774    *    4.584   *   19.061    *   23.645    *
*  17 - 18 *   3.098   *   24.346   *   51.618    *   75.964    *
*  18 - 19 *   2.078   *    9.843   *   15.778    *   25.621    *
*  19 - 20 *   3.288   *   18.197   *   47.877    *   66.074    *
*  20 - 21 *   2.175   *   11.856   *   29.098    *   40.954    *
*  21 - 22 *   4.621   *   28.257   *   76.355    *  104.612    *
*  22 - 23 *   3.857   *   23.956   *   73.651    *   97.607    *
*  23 - 24 *   1.613   *   35.589   *  255.685    *  291.274    *
*-----*
*   0 - 1  *   1.994   *   54.409   *  400.103    *  454.512    *
*   1 - 2  *   1.866   *   98.917   *  768.300    *  867.217    *
*   2 - 3  *   2.931   *   79.285   *  537.762    *  617.047    *
*   3 - 4  *   1.539   *   18.801   *  115.161    *   133.962    *
*   4 - 5  *   2.583   *   10.545   *   19.187    *    29.732    *
*   5 - 6  *   4.580   *   19.997   *   46.309    *    66.306    *
*   6 - 7  *   3.538   *   30.807   *  201.528    *  232.335    *
*   7 - 8  *   2.927   *   29.865   *  213.518    *  243.383    *
*   8 - 9  *   3.405   *   18.411   *   82.049    *   100.460    *
*   9 - 10 *   2.915   *    5.537   *   13.765    *    19.302    *
*  10 - 11 *   2.203   *    4.875   *    6.014    *    10.889    *
*  11 - 12 *   1.320   *    7.215   *   12.571    *    19.786    *
*  12 - 13 *    989    *    5.746   *   14.825    *    20.571    *
*  13 - 14 *    710    *    3.590   *   15.249    *    18.839    *
*  14 - 15 *    752    *    2.772   *    5.855    *     8.627    *
*  15 - 16 *    852    *    3.125   *    6.004    *     9.129    *
*=====*
*  Total   *  55.834   *  564.186   *  3.076.040  *  3.640.226  *
*****

```



## 3) Type d'application 'Front-ends de test' :

```

*****
*   Heure   *   Nombre   *   Segments   *   Segments   *   Total   *
*           *   'calls'  *   reçus    *   envoyés    *   Segments *
*****
*  14 - 15  *    273   *      21    *      42      *    63      *
*  15 - 16  *    493   *    2.040   *    16.192    *   18.232   *
*  16 - 17  *    494   *    541     *    159       *    700      *
*  17 - 18  *    503   *     60     *    108       *    168      *
*  18 - 19  *    496   *     48     *     96       *    144      *
*  19 - 20  *    492   *     48     *     96       *    144      *
*  20 - 21  *    490   *     48     *     96       *    144      *
*  21 - 22  *    490   *     48     *     96       *    144      *
*  22 - 23  *    506   *     90     *    144       *    234      *
*  23 - 24  *    520   *     72     *    144       *    216      *
*-----*
*   0 - 1   *    528   *    393     *    150       *    543      *
*   1 - 2   *    532   *     48     *     96       *    144      *
*   2 - 3   *    524   *     48     *     96       *    144      *
*   3 - 4   *    526   *     48     *     96       *    144      *
*   4 - 5   *    524   *     48     *     96       *    144      *
*   5 - 6   *    528   *     58     *    106       *    164      *
*   6 - 7   *    528   *     48     *     96       *    144      *
*   7 - 8   *    531   *     48     *     96       *    144      *
*   8 - 9   *    208   *     46     *     71       *    117      *
*   9 - 10  *     0    *     0      *     0        *     0       *
*  10 - 11  *    152   *     40     *     55       *     95      *
*  11 - 12  *    471   *     84     *    170       *    254      *
*  12 - 13  *    466   *    104     *    199       *    303      *
*  13 - 14  *    478   *   10.217   *    2.666     *   12.883   *
*  14 - 15  *    487   *     161    *     335      *    496      *
*  15 - 16  *    382   *     266    *     538      *    804      *
*=====*
*  Total   *  11.622  *   14.673   *   22.039    *   36.712   *
*****

```



## 4) Type d'application 'bureautique' :

* Heure *	* Nombre * 'calls' *	* Segments * reçus *	* Segments * envoyés *	* Total * Segments *
* 11 - 12 *	2	192	1.740	1.932
* 12 - 13 *	0	0	0	0
* 13 - 14 *	0	0	0	0
* 14 - 15 *	2	658	2.595	3.253
* 15 - 16 *	1	222	898	1.120
* 16 - 17 *	0	0	0	0
* 17 - 18 *	2	9.052	29.449	38.501
* 18 - 19 *	15	36.518	123.880	160.398
* 19 - 20 *	12	10.067	33.625	43.692
* 20 - 21 *	6	76.967	249.865	326.832
* 21 - 22 *	3	39.621	129.847	169.468
* 22 - 23 *	3	35.964	113.451	149.415
* 23 - 24 *	0	0	0	0
-----				
* 0 - 1 *	3	3.087	9.953	13.040
* 1 - 2 *	2	26.342	83.090	109.432
* 2 - 3 *	0	0	0	0
* 3 - 4 *	0	0	0	0
* 4 - 5 *	0	0	0	0
* 5 - 6 *	0	0	0	0
* 6 - 7 *	0	0	0	0
* 7 - 8 *	0	0	0	0
* 8 - 9 *	1	80	275	355
* 9 - 10 *	5	1.103	734	1.837
* 10 - 11 *	2	3.081	742	3.823
* 11 - 12 *	0	0	0	0
* 12 - 13 *	1	222	898	1.120
=====				
* Total *	60	243.176	781.042	1.024.218

Segments répartis pour le type d'application 'bureautique':

* Heure *	* Nombre *	* Segments *	* Segments *	* Total *
* 'calls' *	* reçus *	* envoyés *	* Segments *	
*****	*****	*****	*****	*****
* 11 - 12 *	2	* 27	* 180	* 207 *
* 12 - 13 *	0	* 27	* 180	* 207 *
* 13 - 14 *	0	* 27	* 180	* 207 *
* 14 - 15 *	2	* 685	* 2.775	* 3.460 *
* 15 - 16 *	1	* 249	* 1.078	* 1.327 *
* 16 - 17 *	0	* 27	* 180	* 207 *
* 17 - 18 *	2	* 3.045	* 9.997	* 13.042 *
* 18 - 19 *	15	* 23.326	* 79.122	* 102.448 *
* 19 - 20 *	12	* 29.294	* 98.175	* 127.469 *
* 20 - 21 *	6	* 28.130	* 90.920	* 119.050 *
* 21 - 22 *	3	* 33.852	* 109.878	* 143.730 *
* 22 - 23 *	3	* 31.188	* 101.108	* 132.296 *
* 23 - 24 *	0	* 22.867	* 74.931	* 97.798 *
-----	-----	-----	-----	-----
* 0 - 1 *	3	* 20.496	* 66.690	* 87.186 *
* 1 - 2 *	2	* 11.999	* 37.983	* 49.982 *
* 2 - 3 *	0	* 10.457	* 33.009	* 43.466 *
* 3 - 4 *	0	* 8.568	* 26.849	* 35.417 *
* 4 - 5 *	0	* 8.568	* 26.849	* 35.417 *
* 5 - 6 *	0	* 2.927	* 9.156	* 12.083 *
* 6 - 7 *	0	* 2.927	* 9.156	* 12.083 *
* 7 - 8 *	0	* 0	* 0	* 0 *
* 8 - 9 *	1	* 27	* 92	* 119 *
* 9 - 10 *	5	* 1.130	* 826	* 1.956 *
* 10 - 11 *	2	* 3.108	* 834	* 3.942 *
* 11 - 12 *	0	* 0	* 0	* 0 *
* 12 - 13 *	1	* 222	* 898	* 1.120 *
=====	=====	=====	=====	=====
* Total *	60	* 243.173	* 781.046	* 1.024.219 *
*****	*****	*****	*****	*****

## 5) Type d'application 'Rj2test' :

```

*****
*  Heure      *  Nombre  *  Segments  *  Segments  *  Total      *
*             *  'calls' *  reçus    *  envoyés   *  Segments   *
*****
*  14 - 15 *    19    *    586    *    3.719   *    4.305    *
*  15 - 16 *    14    *    236    *    692     *    928      *
*  16 - 17 *     8     *    171    *    741     *    912      *
*  17 - 18 *     3     *     58    *    59      *    117      *
*  18 - 19 *     4     *   1.708   *   6.506    *   8.214     *
*  19 - 20 *    11     *   1.357   *   5.146    *   6.503     *
*  20 - 21 *    20     *     40    *     55     *     95      *
*  21 - 22 *     0     *     0     *     0      *     0       *
*  22 - 23 *     2     *     12    *     10     *     22      *
*  23 - 24 *     0     *     0     *     0      *     0       *
*-----*
*  0 - 1 *    10    *    534    *    3.914   *    4.448    *
*  1 - 2 *     1     *     5     *     4      *     9       *
*  2 - 3 *     0     *     0     *     0      *     0       *
*  3 - 4 *     0     *     0     *     0      *     0       *
*  4 - 5 *     1     *     5     *     4      *     9       *
*  5 - 6 *     0     *     0     *     0      *     0       *
*  6 - 7 *     0     *     0     *     0      *     0       *
*  7 - 8 *     0     *     0     *     0      *     0       *
*  8 - 9 *    28     *     6     *     6      *    12       *
*  9 - 10 *   98     *    57     *    18      *    75       *
* 10 - 11 *   83     *   235     *   390      *   625      *
* 11 - 12 *   22     *   205     *   194      *   399      *
* 12 - 13 *   33     *   272     *   215      *   487      *
* 13 - 14 *    4     *    78     *   428      *   506      *
* 14 - 15 *   19     *   198     *   326      *   524      *
* 15 - 16 *   38     *   555     *   1.100    *   1.655    *
*=====*
*  Total   *   418    *   6.318   *   23.527   *   29.845    *
*****

```



## 6) Type d'application 'Call01' :

* Heure *	* Nombre * 'calls' *	* Segments * reçus *	* Segments * envoyés *	* Total * Segments *
* 2 - 3 *	0	0	50	50
* 3 - 4 *	0	0	25	25
* 4 - 5 *	0	0	0	0
* 5 - 6 *	0	165.471	11.251	176.722
* 6 - 7 *	0	2.299	0	2.299
* 7 - 8 *	0	21.022	806	21.828
* 8 - 9 *	1	2.719	792	3.511
* 9 - 10 *	2	2.772	142	2.914
* 10 - 11 *	2	0	0	0
* 11 - 12 *	1	10	6	16
* 12 - 13 *	3	1.697	874	2.571
* 13 - 14 *	1	8	173	181
* 14 - 15 *	11	9.067	26.528	35.595
* 15 - 16 *	13	13.190	39.263	52.453
* 16 - 17 *	23	9.989	28.699	38.688
* 17 - 18 *	12	150.259	54.636	204.895
* 18 - 19 *	20	14.430	42.566	56.996
* 19 - 20 *	4	15.543	43.359	58.902
* 20 - 21 *	11	40.979	31.789	72.768
* 21 - 22 *	17	15.249	39.491	54.740
* 22 - 23 *	20	14.999	44.376	59.375
* 23 - 24 *	4	15.741	47.134	62.875
-----				
* 0 - 1 *	11	10.547	29.910	40.457
* 1 - 2 *	26	14.936	44.271	59.207
* 2 - 3 *	5	14.852	44.401	59.253
* 3 - 4 *	9	10.238	30.433	40.671
* 4 - 5 *	20	15.514	45.951	61.465
* 5 - 6 *	12	14.192	41.910	56.102
* 6 - 7 *	10	10.989	32.655	43.644
* 7 - 8 *	15	15.293	45.716	61.009
* 8 - 9 *	43	13.542	37.862	51.404
* 9 - 10 *	26	8.563	25.002	33.565
* 10 - 11 *	34	12.242	35.536	47.778
* 11 - 12 *	21	8.414	24.542	32.956
* 12 - 13 *	26	13.743	39.995	53.738
* 13 - 14 *	19	9.823	28.541	38.364
* 14 - 15 *	14	6.154	17.309	23.463
* 15 - 16 *	16	373	69	442
=====				
* Total *	452	674.859	936.063	1.610.922
*****				

Segments répartis pour le type d'application 'Call01' :

```

*****
*   Heure   *   Nombre   *   Segments   *   Segments   *   Total   *
*           *   'calls'  *   reçus    *   envoyés    *   Segments *
*****
*   2 - 3   *       0   *       0     *       4     *       4     *
*   3 - 4   *       0   *       0     *       6     *       6     *
*   4 - 5   *       0   *       0     *       6     *       6     *
*   5 - 6   *       0   *    12.728   *      871    *    13.599   *
*   6 - 7   *       0   *    12.905   *      871    *    13.776   *
*   7 - 8   *       0   *    14.522   *     933     *    15.455   *
*   8 - 9   *       0   *    14.762   *    1.002    *    15.764   *
*   9 - 10  *       0   *    14.975   *    1.012    *    15.987   *
*  10 - 11  *       0   *    14.975   *    1.025    *    16.000   *
*  11 - 12  *       2   *    14.977   *    1.026    *    16.003   *
*  12 - 13  *       0   *    15.172   *    1.096    *    16.268   *
*  13 - 14  *       0   *    15.174   *    1.111    *    16.285   *
*  14 - 15  *       2   *    21.907   *   20.872    *   42.779   *
*  15 - 16  *       1   *    28.679   *   41.172    *   69.851   *
*  16 - 17  *       0   *    24.627   *   28.165    *   52.792   *
*  17 - 18  *       2   *    41.165   *   46.061    *   87.226   *
*  18 - 19  *      15   *    28.168   *   44.289    *   72.457   *
*  19 - 20  *      12   *    27.840   *   42.918    *   70.758   *
*  20 - 21  *       6   *    23.534   *   31.448    *   54.982   *
*  21 - 22  *       3   *    27.431   *   42.475    *   69.906   *
*  22 - 23  *       3   *    27.624   *   43.759    *   71.383   *
*  23 - 24  *       0   *    28.886   *   48.021    *   76.907   *
*-----*
*   0 - 1   *       3   *    23.155   *   30.516    *   53.671   *
*   1 - 2   *       2   *    28.308   *   45.783    *   74.091 *
*   2 - 3   *       0   *    27.674   *   44.401    *   72.075   *
*   3 - 4   *       0   *    23.545   *   31.915    *   55.460   *
*   4 - 5   *       0   *    28.695   *   47.015    *   75.710   *
*   5 - 6   *       0   *    27.513   *   43.734    *   71.247   *
*   6 - 7   *       0   *    12.307   *   31.152    *   43.459   *
*   7 - 8   *       0   *    17.411   *   46.799    *   64.210   *
*   8 - 9   *       1   *    13.920   *   39.343    *   53.263   *
*   9 - 10  *       5   *     8.667   *   24.581    *   33.248   *
*  10 - 11  *       2   *    12.729   *   36.341    *   49.070   *
*  11 - 12  *       0   *     8.370   *   24.241    *   32.611   *
*  12 - 13  *       1   *    13.572   *   39.470    *   53.042   *
*  13 - 14  *       0   *     9.831   *   28.203    *   38.034   *
*  14 - 15  *       0   *     8.571   *   24.504    *   33.075   *
*  15 - 16  *       0   *     544     *    106     *    650     *
*=====*
*   Total   *    452   *   674.863   *   936.247   *  1.611.110 *
*****

```

## 7) Type d'application 'Hqr80' :

```

*****
*   Heure   *   Nombre   *   Segments   *   Segments   *   Total   *
*           *   'calls'  *   reçus    *   envoyés    *   Segments *
*****
*  14 - 15  *      6   *      24   *      13   *      37   *
*  15 - 16  *      7   *    10.028 *    21.026 *    31.054 *
*  16 - 17  *     21   *    12.125 *    45.567 *    57.692 *
*  17 - 18  *      8   *    11.528 *    43.648 *    55.176 *
*  18 - 19  *    111   *     8.142 *    29.064 *    37.206 *
*  19 - 20  *      4   *     2.087 *     8.616 *    10.703 *
*  20 - 21  *      5   *    10.630 *    41.678 *    52.308 *
*  21 - 22  *     12   *    17.606 *    67.711 *    85.317 *
*  22 - 23  *     26   *      796  *      311  *     1.107 *
*  23 - 24  *      0   *      0    *      0    *      0    *
*-----*
*   0 - 1   *      0   *      0    *      0    *      0    *
*   1 - 2   *      4   *      24   *      15   *      39   *
*   2 - 3   *      0   *      0    *      0    *      0    *
*   3 - 4   *      6   *     258   *      99   *     357   *
*   4 - 5   *    345   *     8.650 *     3.337 *    11.987 *
*   5 - 6   *    216   *     5.962 *     2.323 *     8.285 *
*   6 - 7   *      7   *     143   *      69   *     212   *
*   7 - 8   *     16   *      0    *      0    *      0    *
*   8 - 9   *     19   *     453   *     177   *     630   *
*   9 - 10  *      6   *    1.270  *     403   *     1.673 *
*  10 - 11  *      0   *      0    *      0    *      0    *
*  11 - 12  *     12   *     863   *     364   *     1.227 *
*  12 - 13  *      7   *      60   *      35   *      95   *
*  13 - 14  *     23   *     2.381 *     2.685 *     5.066 *
*  14 - 15  *      5   *     318   *     127   *     445   *
*  15 - 16  *     10   *      66   *      37   *     103   *
*=====*
*  Total   *    876   *    93.414 *    267.305 *    360.719 *
*****

```



## 8) Type d'application 'Hqc81' :

Heure	Nombre 'calls'	Segments reçus	Segments envoyés	Total Segments
14 - 15	48	1.671	645	2.316
15 - 16	121	2.013	811	2.824
16 - 17	48	1.319	519	1.838
17 - 18	50	1.061	438	1.499
18 - 19	98	1.164	804	1.968
19 - 20	1.006	7.458	3.855	11.313
20 - 21	292	2.343	1.251	3.594
21 - 22	44	549	239	788
22 - 23	13	300	118	418
23 - 24	0	0	0	0
-----				
0 - 1	152	2.810	2.701	5.511
1 - 2	11	214	107	321
2 - 3	19	133	69	202
3 - 4	130	3.070	1.179	4.249
4 - 5	749	17.234	6.226	23.460
5 - 6	314	5.898	2.156	8.054
6 - 7	8	23	12	35
7 - 8	48	1.076	453	1.529
8 - 9	144	2.576	1.119	3.695
9 - 10	164	8.713	23.431	32.144
10 - 11	216	3.593	1.727	5.320
11 - 12	398	4.515	2.376	6.891
12 - 13	548	11.450	13.286	24.736
13 - 14	731	5.375	2.943	8.318
14 - 15	73	2.806	2.490	5.296
15 - 16	98	5.065	3.370	8.435
=====				
Total	5.523	92.429	72.325	164.754

## 9) Type d'application 'Salles de cours' :

Heure	Nombre 'calls'	Segments reçus	Segments envoyés	Total Segments
14 - 15	84	575	653	1.228
15 - 16	141	1.047	1.159	2.206
16 - 17	125	907	810	1.717
17 - 18	125	802	654	1.456
18 - 19	129	778	631	1.409
19 - 20	137	789	670	1.459
20 - 21	142	742	660	1.402
21 - 22	147	780	694	1.474
22 - 23	142	783	719	1.502
23 - 24	43	132	156	288
-----				
0 - 1	81	535	573	1.108
1 - 2	48	167	187	354
2 - 3	50	153	182	335
3 - 4	49	145	149	294
4 - 5	55	131	142	273
5 - 6	50	169	190	359
6 - 7	32	173	166	339
7 - 8	18	92	137	229
8 - 9	52	816	889	1.705
9 - 10	146	835	946	1.781
10 - 11	170	853	875	1.728
11 - 12	145	925	1.811	2.736
12 - 13	128	767	666	1.433
13 - 14	134	740	707	1.447
14 - 15	150	866	856	1.722
15 - 16	121	815	950	1.765
=====				
<b>Total</b>	<b>2.644</b>	<b>15.517</b>	<b>16.232</b>	<b>31.749</b>



10) Type d'application 'Interccer' :

```

*****
*  Heure      *  Nombre  *  Segments  *  Segments  *  Total      *
*            *  'calls' *  reçus    *  envoyés   *  Segments   *
*****
*   6 - 7    *      2   *      0    *      0     *      0       *
*  11 - 12   *      1   *  49.831  *  21.932   *  71.763     *
*  18 - 19   *      0   *      0    *      0     *      0       *
*  20 - 21   *      0   *  35.888  *  24.765   *  60.653     *
*  23 - 24   *      0   *  35.414  *  16.251   *  51.665     *
*=====
*  Total     *      3   *  121.133 *  62.948   *  184.081    *
*****

```

Segments répartis pour le type d'application 'Interccer' :

```

*****
*  Heure      *  Nombre  *  Segments  *  Segments  *  Total      *
*            *  'calls' *  reçus    *  envoyés   *  Segments   *
*****
*  11 - 12   *      1   *   5.537  *   2.437   *   7.974     *
*  12 - 13   *      0   *   5.537  *   2.437   *   7.974     *
*  13 - 14   *      0   *   5.537  *   2.437   *   7.974     *
*  14 - 15   *      0   *   5.537  *   2.437   *   7.974     *
*  15 - 16   *      0   *   5.537  *   2.437   *   7.974     *
*  16 - 17   *      0   *   5.537  *   2.437   *   7.974     *
*  17 - 18   *      0   *   5.537  *   2.437   *   7.974     *
*  18 - 19   *      0   *   5.537  *   2.437   *   7.974     *
*  19 - 20   *      0   *   5.537  *   2.437   *   7.974     *
*  20 - 21   *      0   *   8.972  *   6.191   *  15.163     *
*  21 - 22   *      0   *   8.972  *   6.191   *  15.163     *
*  22 - 23   *      0   *   8.972  *   6.191   *  15.163     *
*  23 - 24   *      0   *  11.696  *   7.441   *  19.137     *
*-----
*   0 - 1    *      0   *   2.724  *   1.250   *   3.974     *
*   1 - 2    *      0   *   2.724  *   1.250   *   3.974     *
*   2 - 3    *      0   *   2.724  *   1.250   *   3.974     *
*   3 - 4    *      0   *   2.724  *   1.250   *   3.974     *
*   4 - 5    *      0   *   2.724  *   1.250   *   3.974     *
*   5 - 6    *      0   *   2.724  *   1.250   *   3.974     *
*   6 - 7    *      0   *   2.724  *   1.250   *   3.974     *
*   7 - 8    *      0   *   2.724  *   1.250   *   3.974     *
*   8 - 9    *      0   *   2.724  *   1.250   *   3.974     *
*   9 - 10   *      0   *   2.724  *   1.250   *   3.974     *
*  10 - 11   *      0   *   2.724  *   1.250   *   3.974     *
*  11 - 12   *      0   *   2.724  *   1.250   *   3.974     *
*=====
*  Total     *      3   *  121.133 *  62.947   *  184.080     *
*****

```



## 11) Type d'application 'Portefeuille' :

```

*****
*  Heure      *  Nombre  *  Segments  *  Segments  *  Total    *
*              *  'calls' *  reçus    *  envoyés   *  Segments *
*****
* 16 - 17 *      8 *      201 *      1.284 *      1.485 *
* 17 - 18 *     13 *     2.028 *     16.775 *     18.803 *
* 18 - 19 *      3 *       8 *       8 *       16 *
* 19 - 20 *      3 *      31 *      28 *       59 *
* 20 - 21 *      1 *      15 *      12 *       27 *
*-----*
* Total    *     28 *     2.283 *     18.107 *     20.390 *
*****

```

## 12) Type d'application 'Gline' :

```

*****
*  Heure      *  Nombre  *  Segments  *  Segments  *  Total    *
*              *  'calls' *  reçus    *  envoyés   *  Segments *
*****
* 14 - 15 *      3 *       58 *      108 *      166 *
* 15 - 16 *      7 *      124 *       95 *      219 *
* 16 - 17 *      0 *        0 *        0 *        0 *
* 17 - 18 *      5 *      154 *       93 *      247 *
* 18 - 19 *      1 *        5 *        8 *       13 *
* 19 - 20 *      7 *      193 *      114 *      307 *
* 20 - 21 *      0 *        0 *        0 *        0 *
* 21 - 22 *      2 *       29 *       25 *       54 *
* 22 - 23 *      3 *       35 *       68 *      103 *
* 23 - 24 *      4 *       36 *      109 *      145 *
*-----*
* 0 - 1 *      2 *       17 *       49 *       66 *
* 1 - 2 *      0 *        0 *        0 *        0 *
* 2 - 3 *      0 *        0 *        0 *        0 *
* 3 - 4 *      4 *       25 *       36 *       61 *
* 4 - 5 *      0 *        0 *        0 *        0 *
* 5 - 6 *      0 *        0 *        0 *        0 *
* 6 - 7 *      2 *        1 *        2 *        3 *
* 7 - 8 *      5 *       62 *       52 *      114 *
* 8 - 9 *      4 *       44 *       83 *      127 *
* 9 - 10 *      3 *       34 *       40 *       74 *
* 10 - 11 *      6 *      161 *       96 *      257 *
* 11 - 12 *      6 *      111 *       52 *      163 *
* 12 - 13 *      3 *       36 *       52 *       88 *
* 13 - 14 *      2 *       40 *       27 *       67 *
* 14 - 15 *      2 *       19 *       32 *       51 *
* 15 - 16 *      5 *      102 *      116 *      218 *
*-----*
* Total    *     76 *     1.286 *     1.257 *     2.543 *
*****

```

### 13) Type d'application 'Agences' :

Comme nous l'avons déjà signalé, les différents résultats concernant les agences sont obtenus par différence entre les données des fichiers 'SEGREC.DAT', 'SEGSSEN.DAT' et 'CALLNET.DAT', et les données obtenues pour les douze types d'application mentionnés ci-dessus.

Les détails des calculs seront donnés pour la première journée uniquement; pour les autres journées, seuls les tableaux seront fournis dans l'annexe page XX.

#### Détail des calculs :

##### a) le nombre de segments :

Ils sont obtenus de la manière suivante :

Pour une heure déterminée :

total des segments reçus et envoyés (fichiers 'SEGREC.DAT' et  
'SEGSSEN.DAT')

- total des segments pour 'CCNC'
- total des segments pour les 'FRONT-ENDS'
- total des segments pour les 'FRONT-ENDS DE TEST'
- total des segments pour la 'BUREAUTIQUE'
- total des segments pour 'RJ2TEST'
- total des segments pour 'HQR80'
- total des segments pour 'HQC81'
- total des segments pour les 'SALLES DE COURS'
- total des segments pour 'INTERCER'
- total des segments pour le 'PORTEFEUILLE'
- total des segments pour 'GLINE'

-----  
= total des segments pour les 'AGENCES'

Pratiquement, pour la journée considérée, cela nous donne :

17 - 18 heures :	215.791	-	191.931	=	23.860 segments.
18 - 19 heures :	277.103	-	234.608	=	42.495 segments.
19 - 20 heures :	161.196	-	140.254	=	20.942 segments.
20 - 21 heures :	496.568	-	425.356	=	71.212 segments.
21 - 22 heures :	384.092	-	361.857	=	22.235 segments.
22 - 23 heures :	266.757	-	250.408	=	16.349 segments.
23 - 24 heures :	349.672	-	343.588	=	6.084 segments.

---

0 - 1 heure :	518.440	-	479.228	=	39.212 segments.
1 - 2 heures :	993.474	-	977.516	=	15.958 segments.
2 - 3 heures :	638.387	-	617.728	=	20.659 segments.
3 - 4 heures :	151.414	-	139.076	=	12.338 segments.
4 - 5 heures :	76.060	-	65.890	=	10.170 segments.
5 - 6 heures :	96.290	-	83.168	=	13.122 segments.
6 - 7 heures :	238.055	-	233.068	=	4.987 segments.
7 - 8 heures :	250.232	-	245.399	=	4.833 segments.
8 - 9 heures :	226.596	-	107.101	=	119.495 segments.
9 - 10 heures :	313.963	-	56.886	=	257.077 segments.
10 - 11 heures :	321.722	-	28.702	=	293.020 segments.
11 - 12 heures :	385.450	-	31.456	=	353.994 segments.
12 - 13 heures :	336.343	-	48.836	=	287.507 segments.
13 - 14 heures :	169.074	-	47.126	=	121.948 segments.
14 - 15 heures :	259.900	-	17.161	=	242.739 segments.
15 - 16 heures :	258.253	-	22.109	=	236.144 segments.

**Total :** **2.236.380 segments.**



b) le nombre de 'calls' :

La méthode utilisée est la même que celle décrite au point précédent; les résultats sont les suivants :

17 - 18 heures :	5.224 - 3.807 =	1.417 'calls'.
18 - 19 heures :	4.285 - 2.938 =	1.347 'calls'.
19 - 20 heures :	6.772 - 4.960 =	1.812 'calls'.
20 - 21 heures :	4.041 - 3.131 =	910 'calls'.
21 - 22 heures :	7.482 - 5.319 =	2.163 'calls'.
22 - 23 heures :	6.118 - 4.552 =	1.566 'calls'.
23 - 24 heures :	2.788 - 2.180 =	608 'calls'.

---

0 - 1 heure :	5.176 - 2.770 =	2.406 'calls'.
1 - 2 heures :	3.479 - 2.464 =	1.015 'calls'.
2 - 3 heures :	5.146 - 3.524 =	1.622 'calls'.
3 - 4 heures :	3.228 - 2.254 =	974 'calls'.
4 - 5 heures :	5.314 - 4.257 =	1.057 'calls'.
5 - 6 heures :	6.924 - 5.688 =	1.236 'calls'.
6 - 7 heures :	4.617 - 4.115 =	502 'calls'.
7 - 8 heures :	3.931 - 3.545 =	386 'calls'.
8 - 9 heures :	7.253 - 3.861 =	3.392 'calls'.
9 - 10 heures :	8.386 - 3.337 =	5.049 'calls'.
10 - 11 heures :	9.821 - 2.836 =	6.985 'calls'.
11 - 12 heures :	8.538 - 2.374 =	6.164 'calls'.
12 - 13 heures :	7.405 - 2.175 =	5.230 'calls'.
13 - 14 heures :	5.061 - 2.082 =	2.979 'calls'.
14 - 15 heures :	6.597 - 1.488 =	5.109 'calls'.
15 - 16 heures :	5.509 - 1.506 =	4.003 'calls'.

**Total :** **57.932 'calls'.**

### 3.2.2 Comportement horaire pour l'ensemble du réseau.

Nous développerons les calculs pour la première journée uniquement; pour les autres journées, seuls les tableaux 'résultats' seront fournis (voir pages 101 et 102).

Suite aux remarques concernant les segments répartis pour les types d'application 'Call01', 'bureautique' et 'intercer', les différents résultats seront obtenus de la manière suivante :

Journée du 15/16 octobre 1985 :

Soit une heure déterminée,

total des segments reçus et envoyés (fichiers 'SEGREC. DAT'	
et 'SEGSEN. DAT')	
- total des segments pour la bureautique	
+ total des segments répartis pour la bureautique	
- total des segments pour 'intercer'	
+ total des segments répartis pour 'intercer'	
+ total des segments répartis pour les 'Call01'	
-----	
= total des segments qui transitent sur le réseau pour	
l'heure spécifiée.	

De manière numérique, cela nous donne :

17 - 18 heures :	215.791	
	- 38.501 + 13.042	
	- 0 + 7.974	
	+ 87.226	= 285.532 segments.
18 - 19 heures :	277.103	
	- 160.398 + 102.448	
	- 0 + 7.974	
	+ 72.457	= 299.584 segments.
19 - 20 heures :	161.196	
	- 43.692 + 127.469	
	- 0 + 7.974	
	+ 70.758	= 323.705 segments.
20 - 21 heures :	496.568	
	- 326.832 + 119.050	
	- 60.653 + 15.163	
	+ 54.982	= 298.278 segments.
21 - 22 heures :	384.092	
	- 169.468 + 143.730	
	- 0 + 15.163	
	+ 69.906	= 443.423 segments.
22 - 23 heures :	266.757	
	- 149.415 + 132.296	
	- 0 + 15.163	
	+ 71.383	= 336.184 segments.

23 - 24 heures : 349.672  
 - 0 + 97.798  
 - 51.665 + 19.137  
 + 76.907 = 491.849 segments.

0 - 1 heure : 518.440  
 - 13.040 + 87.186  
 - 0 + 3.974  
 + 53.671 = 650.231 segments.

1 - 2 heures : 993.474  
 - 109.432 + 49.982  
 - 0 + 3.974  
 + 74.091 = 1.012.089 segments.

2 - 3 heures : 638.387  
 - 0 + 43.466  
 - 0 + 3.974  
 + 72.075 = 757.902 segments.

3 - 4 heures : 151.414  
 - 0 + 35.417  
 - 0 + 3.974  
 + 55.460 + 6 = 246.271 segments.

4 - 5 heures : 76.060  
 - 0 + 35.417  
 - 0 + 3.974  
 + 75.710 + 6 = 191.167 segments.

5 - 6 heures : 96.290  
 - 0 + 12.083  
 - 0 + 3.974  
 + 71.247 + 14.833 = 198.427 segments.

6 - 7 heures : 238.055  
 - 0 + 12.083  
 - 0 + 3.974 + 2  
 + 43.459 + 15.040 = 312.613 segments.

7 - 8 heures : 250.232  
 - 0 + 0  
 - 0 + 3.974 + 2  
 + 64.210 + 16.761 = 335.179 segments.

8 - 9 heures : 226.596  
 - 355 + 119  
 - 0 + 3.974 + 2  
 + 53.263 + 16.975 = 300.574 segments.

9 - 10 heures : 313.963  
 - 1.837 + 1.956  
 - 0 + 3.974 + 2  
 + 33.248 + 16.975 = 368.281 segments.



10 - 11 heures : 321.722  
- 3.823 + 3.942  
- 0 + 3.974 + 2  
+ 49.070 + 17.223 = 392.110 segments.

11 - 12 heures : 385.450  
- 0 + 0  
- 0 + 3.974 + 9.483  
+ 32.611 + 17.226 = 448.744 segments.

12 - 13 heures : 336.343  
- 1.120 + 1.120  
- 0 + 9.483  
+ 53.042 + 17.336 = 416.204 segments.

13 - 14 heures : 169.074  
- 0 + 0  
- 0 + 9.483  
+ 38.034 + 17.355 = 233.946 segments.

14 - 15 heures : 259.900  
- 0 + 0  
- 0 + 9.483  
+ 33.075 + 17.362 = 319.820 segments.

15 - 16 heures : 258.255  
- 0 + 0  
- 0 + 9.483  
+ 650 + 26.154 = 294.540 segments.

Les résultats des autres journées sont repris dans les tableaux ci-après :

*****				
* Heure *	16/17	* 21/22	* 22/23	*
*****				
* 17 - 18 *	275.737	* 198.926	* 164.610	*
* 18 - 19 *	243.234	* 286.266	* 243.620	*
* 19 - 20 *	309.632	* 291.583	* 365.134	*
* 20 - 21 *	276.894	* 240.400	* 430.984	*
* 21 - 22 *	407.977	* 319.044	* 364.593	*
* 22 - 23 *	293.200	* 431.845	* 342.395	*
* 23 - 24 *	310.996	* 480.547	* 558.452	*
* - - - - - *				
* 0 - 1 *	519.384	* 597.940	* 614.506	*
* 1 - 2 *	742.024	* 550.423	* 675.032	*
* 2 - 3 *	314.619	* 764.263	* 437.230	*
* 3 - 4 *	121.723	* 292.129	* 360.847	*
* 4 - 5 *	209.593	* 228.740	* 86.188	*
* 5 - 6 *	235.905	* 153.305	* 69.261	*
* 6 - 7 *	76.733	* 86.030	* 42.117	*
* 7 - 8 *	53.429	* 97.727	* 49.300	*
* 8 - 9 *	179.517	* 229.253	* 168.945	*
* 9 - 10 *	369.349	* 401.938	* 366.057	*
* 10 - 11 *	434.119	* 446.647	* 387.126	*
* 11 - 12 *	423.702	* 471.870	* 423.215	*
* 12 - 13 *	335.438	* 390.153	* 385.330	*
* 13 - 14 *	270.725	* 225.549	* 218.414	*
* 14 - 15 *	386.222	* 350.075	* 331.056	*
* 15 - 16 *	406.854	* 452.489	* 421.225	*
* 16 - 17 *	294.332	* 379.602	* 287.128	*
*****				

*****				
* Heure *	26/27	*	30/31	* 31/10 1/11 *
*****				
* 17 - 18 *	-	*	174.189	* 589.459 *
* 18 - 19 *	12.776	*	342.254	* 632.216 *
* 19 - 20 *	29.458	*	352.343	* 355.616 *
* 20 - 21 *	29.829	*	334.057	* 572.921 *
* 21 - 22 *	29.094	*	327.290	* 507.018 *
* 22 - 23 *	29.653	*	382.028	* 506.734 *
* 23 - 24 *	27.798	*	427.393	* 593.077 *
*-----*				
* 0 - 1 *	34.109	*	441.381	* 771.516 *
* 1 - 2 *	28.253	*	560.194	* 880.922 *
* 2 - 3 *	27.143	*	628.579	* 1.020.330 *
* 3 - 4 *	27.431	*	505.212	* 611.864 *
* 4 - 5 *	27.425	*	345.032	* 538.103 *
* 5 - 6 *	24.452	*	175.460	* 553.463 *
* 6 - 7 *	29.616	*	97.542	* 350.744 *
* 7 - 8 *	25.685	*	86.952	* 254.630 *
* 8 - 9 *	28.872	*	169.229	* 148.434 *
* 9 - 10 *	27.653	*	385.743	* 311.628 *
* 10 - 11 *	27.192	*	472.605	* 198.333 *
* 11 - 12 *	27.348	*	484.253	* 25.375 *
* 12 - 13 *	20.519	*	707.295	* 20.849 *
* 13 - 14 *	23.422	*	681.869	* 114.626 *
* 14 - 15 *	19.659	*	948.207	* 157.712 *
* 15 - 16 *	22.978	*	750.338	* 121.427 *
* 16 - 17 *	23.053	*	589.481	* 241.426 *
*****				

#### Remarque :

Il a fallu tenir compte, pour chaque jour, des recouvrements éventuels pouvant survenir entre deux fichiers 'résultats' (c'est notamment le cas lorsque nous avons exploité les différentes statistiques de deux jours consécutifs).

#### 3.2.3 Graphiques obtenus.

Avec les différents résultats obtenus aux différents paragraphes de ce chapitre, nous avons établi trois types de graphiques :

- 1) un type de graphique '**TRAFIC**' par type d'application. Ce type de graphique reprend par tranche horaire et par type d'application, le nombre de segments transitant sur le réseau. Nous n'avons retenu que les types d'application significatifs.
- 2) un type de graphique '**CALLS**' par type d'application. Même remarque que pour le point précédent.
- 3) un type de graphique '**SEGMENTS**' pour l'ensemble du réseau. Ce graphique reprend par tranche horaire et pour l'ensemble du réseau, le nombre de segments transitant sur le réseau.



Pour la journée du 15/16 octobre 1985, les graphiques sont les suivants : (les graphiques concernant les autres journées sont repris en annexe pages 112 et suivantes)

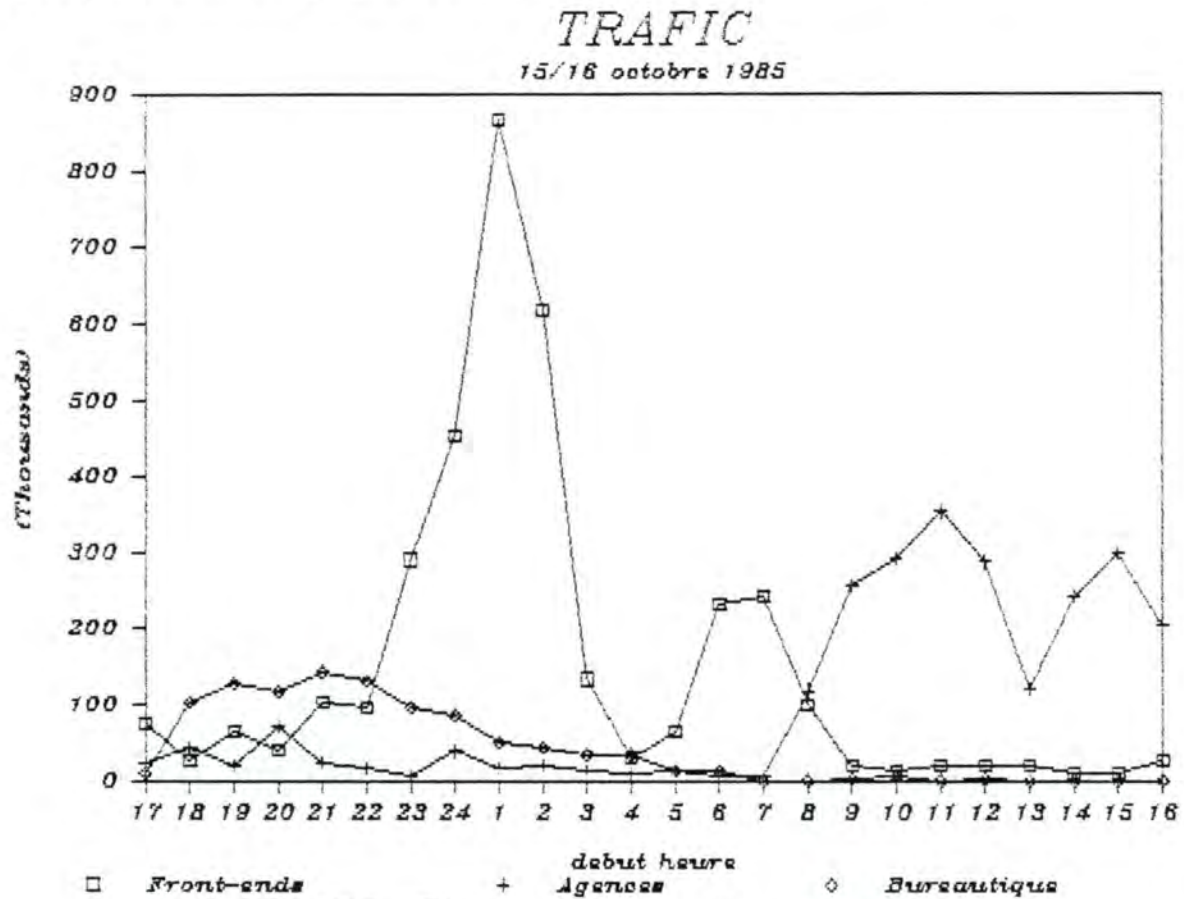


Figure 3.1 : Type de graphique 'TRAFIC'.

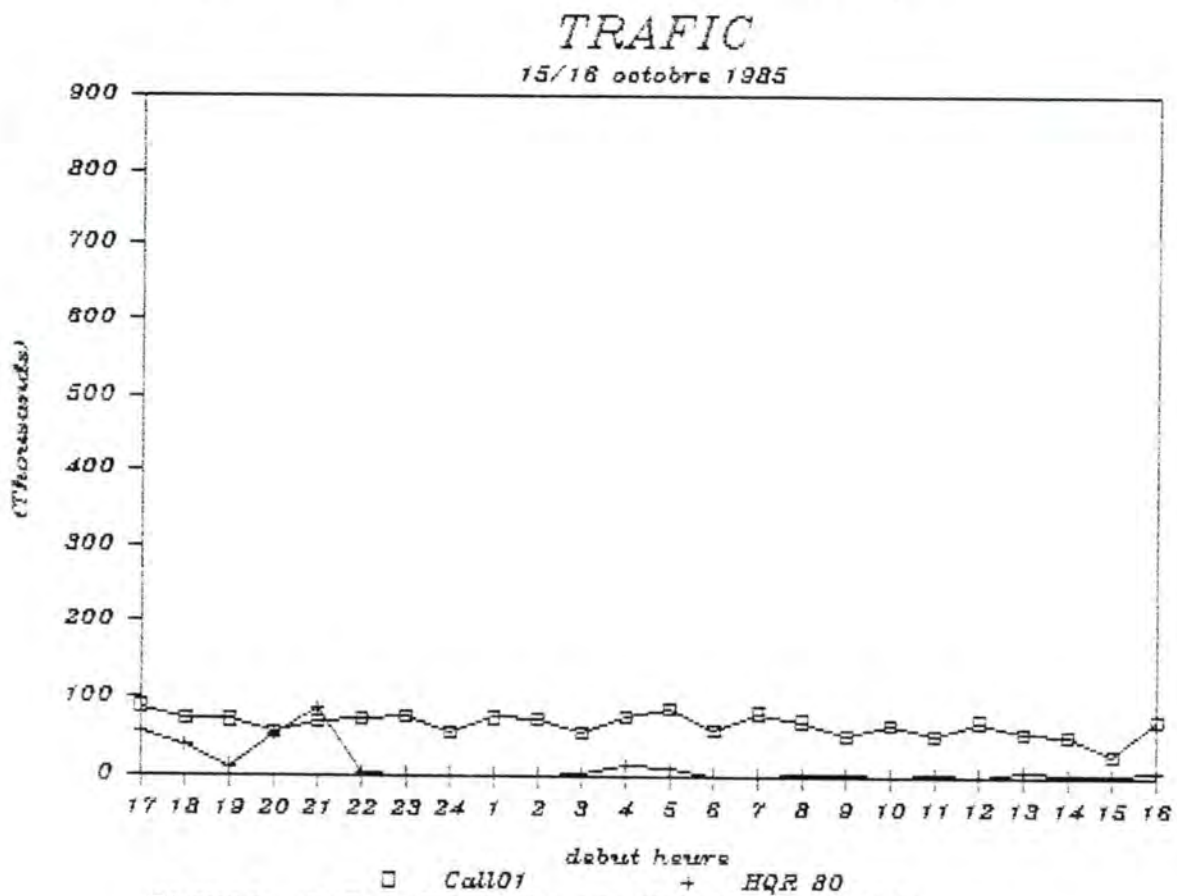


Figure 3.2 : Type de graphique 'TRAFIC'.

# GRAPHIQUE CALLS

15/16 OCTOBRE 1985

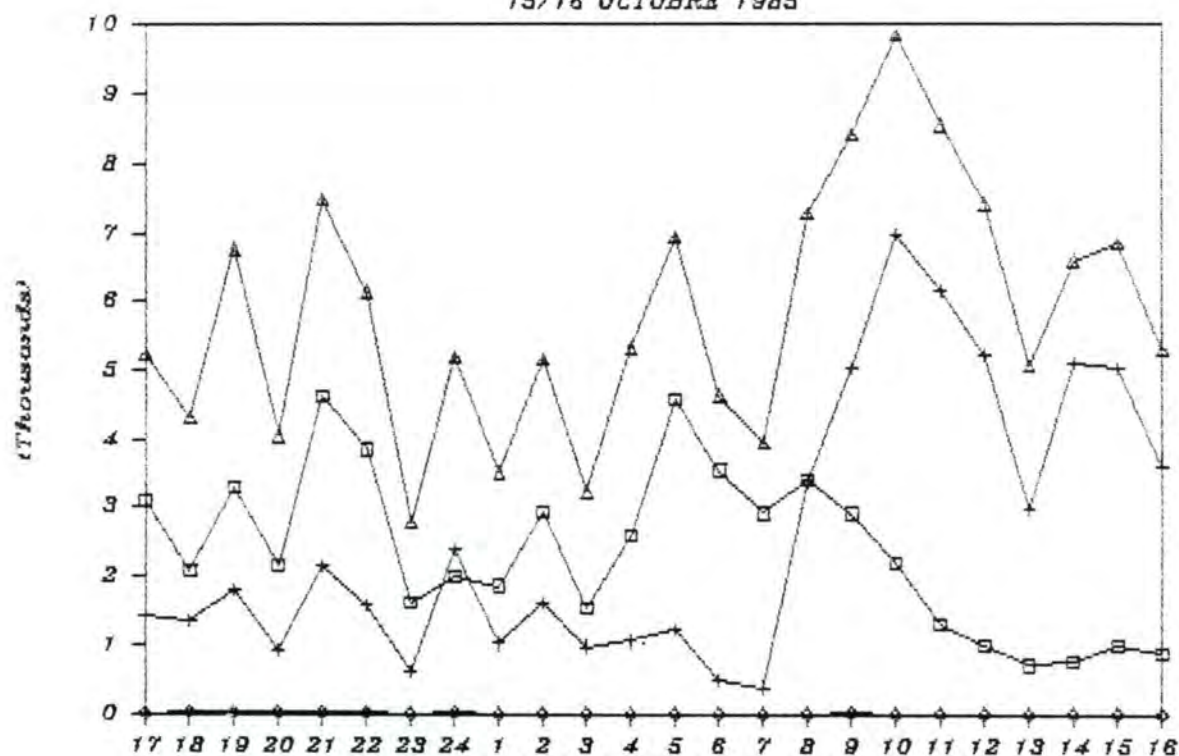


Figure 3.3 : Type de graphique 'CALLS'.

# GRAPHIQUE CALLS

15/16 OCTOBRE 1985

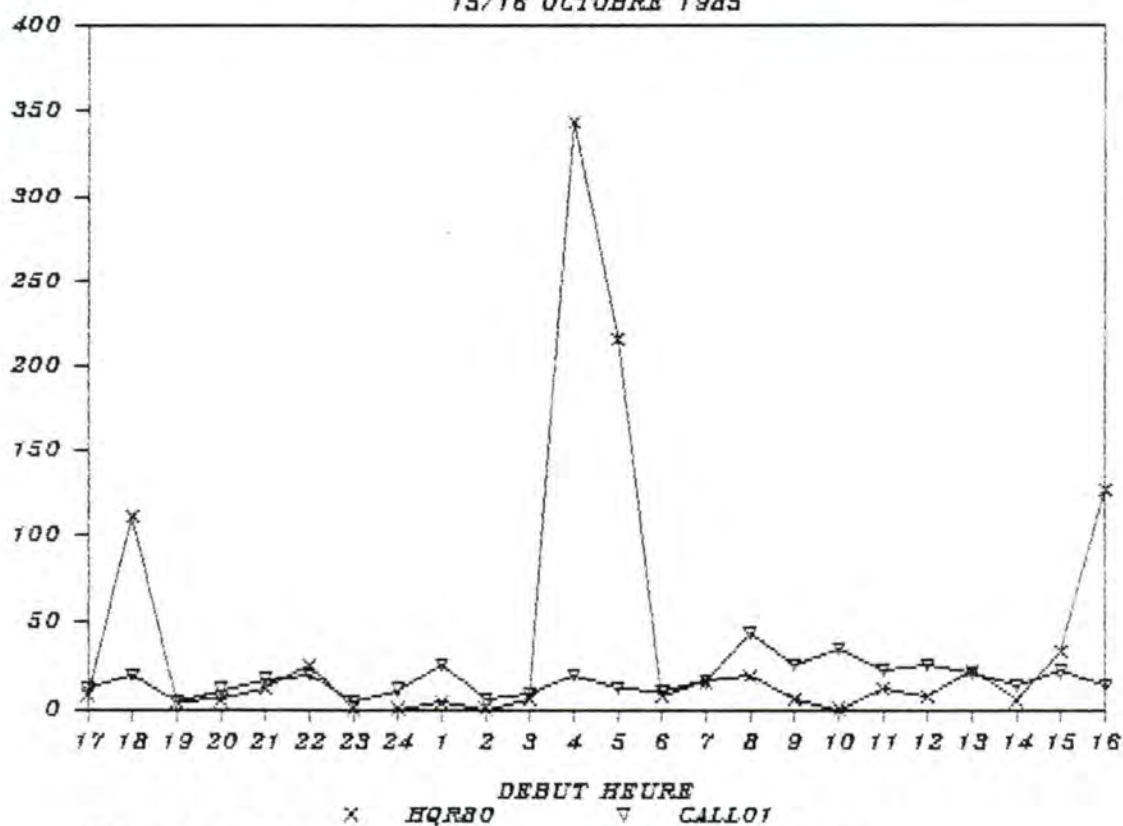
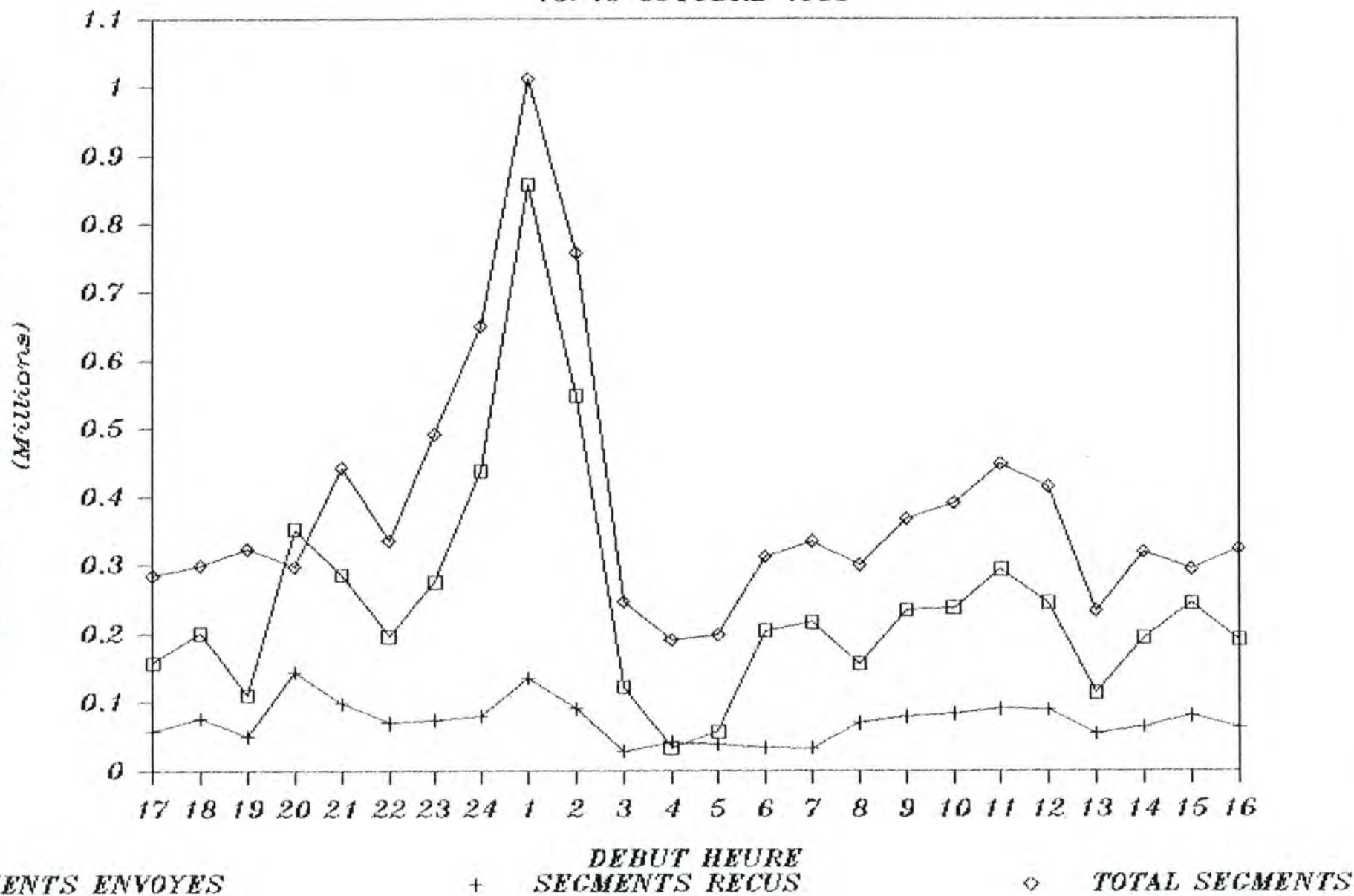


Figure 3.4 : Type de graphique 'CALLS'.

# GRAPHIQUE SEGMENTS

15/16 OCTOBRE 1985

Figure 3.5 : Type de graphique 'SEGMENTS'.





### 3.3 CONCLUSIONS.

En examinant les différents graphiques, nous constatons que le type d'application 'CALLO1' donne au réseau une charge faible et uniforme (inférieure à 85.000 segments par heure). De ce fait, nous avons décidé de ne plus en tenir compte dans la suite de notre travail.

Par contre, pour les types d'application 'FRONT-ENDS', 'AGENCES' et 'BUREAUTIQUE', il n'en va pas de même : non seulement, les charges sont plus importantes, mais elles sont également fonction de la période de la journée à laquelle la mesure est effectuée.

Comme nous avons décidé, d'étudier le comportement du réseau aux périodes critiques (c'est-à-dire les plus chargées), nous avons choisi d'étudier uniquement le comportement des trois types d'application mentionnés ci-dessus; ceci signifie que nous obtiendrons des valeurs pour les variables citées au chapitre deux, uniquement pour chacun de ces trois derniers types d'application et aux heures les plus chargées de la journée.

#### **Quelles sont ces heures ?**

En général, pour les 'FRONT-ENDS' : entre 23 h et 3 h.  
 pour les 'AGENCES' : entre 9 h et 12 h.  
 pour la 'BUREAUTIQUE' : entre 18 h et 22 h.

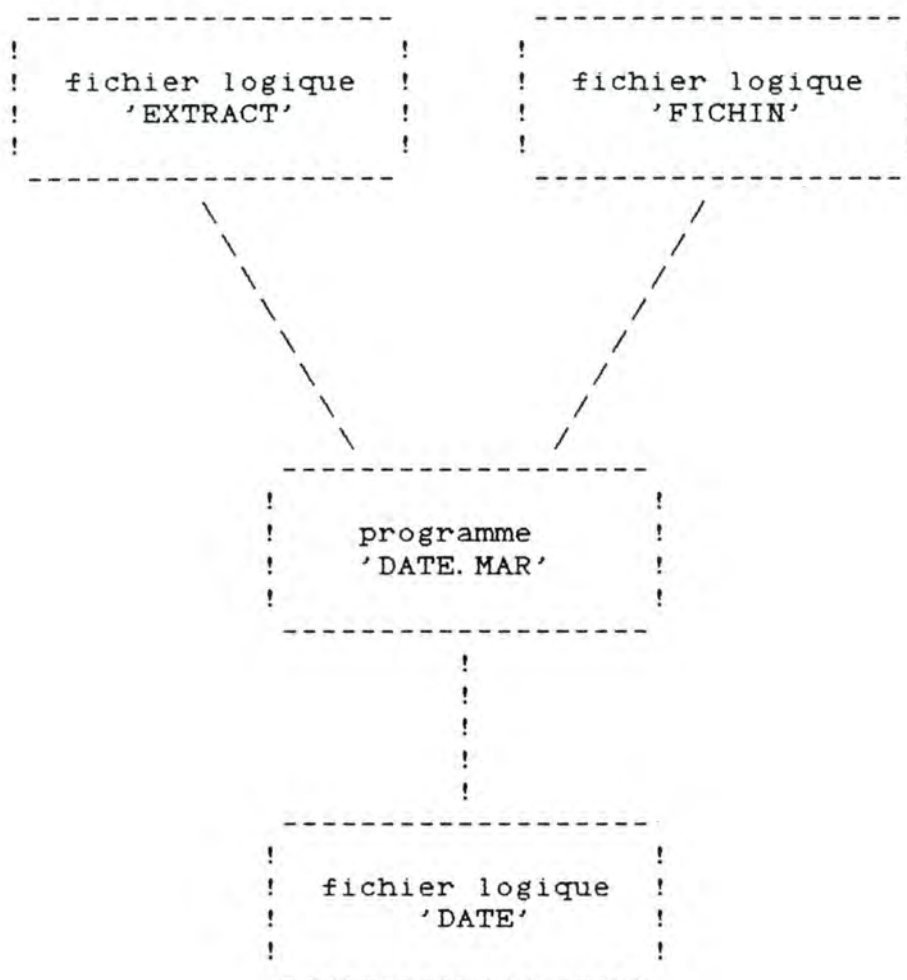
Après avoir décidé de ne retenir que les périodes de charge maximale pour chaque type d'application (front-ends, agences et bureautique), il s'est avéré nécessaire d'écrire une nouvelle procédure qui permet d'extraire des différents fichiers 'AGENCES.DAT', 'FEP.DAT' et 'BUR.DAT' les enregistrements se rapportant aux heures spécifiées ci-dessus. Cette procédure s'appelle 'DATE.MAR' et est écrite en MACRO (annexe page 140).

#### Procédure 'DATE.MAR' :

##### Spécification du programme :

Le programme a pour effet d'extraire du fichier logique 'EXTRACT' tous les enregistrements dont la période de l'appel se trouve entièrement comprise entre les deux dates et heures spécifiées dans le fichier 'FICHIN' et de les écrire dans le fichier logique 'DATE'. Les deux dates et heures du fichier 'FICHIN' seront spécifiées à l'aide de quatre enregistrements répartis de la manière suivante :

1<sup>er</sup> enregistrement : H1 = heure au jour J1.  
 2<sup>ème</sup> enregistrement : J1 = date du premier jour.  
 3<sup>ème</sup> enregistrement : H2 = heure au jour J2.  
 4<sup>ème</sup> enregistrement : J2 = date du deuxième jour.

Schéma :Structure des fichiers :*- fichier logique 'EXTRACT' :*

Sa structure est la même que celle du fichier 'ACNT. DAT' décrit à la page 36.

*- fichier logique 'FICHIN' :*

Il s'agit d'un fichier séquentiel comportant quatre enregistrements d'une longueur de deux caractères chacun.

*- fichier logique 'DATE' :*

Sa structure est la même que celle du fichier 'ACNT. DAT' décrit à la page 37.



Est-il raisonnable de ne retenir que ces trois types d'application ?

Pour répondre à cette question, nous pouvons établir le tableau suivant où nous faisons apparaître le pourcentage de charge de chacun des trois types d'application, relativement à la charge totale du réseau (total des segments envoyés et reçus), de même que le pourcentage global pour les trois types d'application, relativement à la charge totale du réseau.

d	total	segments	segments	segments	pourcentage
a	segments	agences	front-ends	bureautique	global
t					
e					
15	7.384.832	2.236.380	3.554.203	1.017.913	
16		30,28 %	48 %	13,78 %	91 %
16	6.426.526	2.667.053	2.292.528	756.390	
17		41 %	35 %	11,76 %	87 %
21	6.318.781	2.268.142	2.589.846	909.202	
22		35,8 %	41 %	14,38 %	91 %
22	6.387.687	2.234.925	2.643.376	912.280	
23		34,98 %	41,38 %	14,28 %	90,64 %
26	90.508	15.526	53.604	0	
27		17,15 %	59,22 %	0 %	76,38 %
30	9.186.036	2.687.515	2.333.587	867.013	
31		31 %	25 %	9 %	65 %
!	!	!	!	!	!
!	! Sans 'HQR80' (1.901.850 seg = 20 % du total) -->				83,29 %
31	9.065.810	306.112	4.598.350	622.716	
01		3,37 %	50 %	6,86 %	60 %
!	!	!	!	!	!
!	! Sans 'HQR80' (3.088.478 seg = 34 % du total) -->				88,90 %

Nous pouvons conclure de ceci que, mis à part les deux dernières journées, et tenant compte du fait que les mesures effectuées du 26 au 27 octobre 1985 étaient celles d'un week-end, le pourcentage global est toujours supérieur à 83 %. Remarquons que ceci se vérifie également pour les deux dernières journées si l'on fait abstraction de la charge fournie par le type d'application 'HQR80' (Durant ces deux dernières journées, il avait été procédé à un transfert de fichiers au départ de la machine reliée au noeud 80). Il n'est donc pas étonnant de ne retenir que les trois types d'application susmentionnés dans notre étude qui suit.



## CHAPITRE 4

### PRISE DES MESURES DES VARIABLES A EXPLIQUER.

-----

#### Résumé :

L'objectif de ce chapitre est d'expliquer les différentes méthodes que nous avons adoptées pour mesurer l'ensemble des variables présentées au chapitre deux.

Pour chaque variable, nous présentons d'abord les problèmes que nous avons rencontrés pour la mesurer (Mesure de cette variable); ensuite, pour chaque type d'application retenu à la suite du chapitre trois, nous développons la façon de l'obtenir concrètement (Prise des mesures); enfin, nous fournissons un ensemble de résultats, que nous utiliserons pour certaines variables lors de l'étude statistique du chapitre cinq (Résultats obtenus).

-----

Remarque :

La variable "MOY" qui se trouve dans tous les tableaux de ce chapitre sert à l'étude statistique dont question au chapitre cinq.

4.1 UTILISATION DES PROCESSEURS DE LIGNES.4.1.1 Mesure de cette variable :

Pour mesurer cette variable, nous avons utilisé le fichier 'LINEPROC.SRT' décrit à la page 40. Grâce à ce fichier, il nous est possible de connaître le pourcentage d'utilisation d'un processeur de lignes déterminé durant une période statistique de plus ou moins quinze minutes. Il suffit de suivre la démarche suivante :

- Repérer dans le fichier l'enregistrement dont les numéros du noeud et du processeur sont identiques à ceux du processeur étudié (OFFICE\_CODE et LP\_NUMBER)
- et qui correspond à la période analysée (TIME).
- Enfin, appliquer la formule suivante afin d'en déduire le taux d'utilisation demandé :

$$\frac{\text{OS\_RUN\_TIME} + (\text{DR\_RUN\_TIME} * (\text{OS\_RUN\_COUNT} / \text{DR\_RUN\_COUNT}))}{\text{DR\_RUN\_TIME} + \text{OS\_RUN\_TIME}} * 100$$

Cette formule s'explique de la manière suivante :

Un processeur de lignes boucle continuellement dans le gestionnaire de processeur (appelé LP\_DRIVER) pour vérifier s'il y a un (des) message(s) à traiter dans sa file d'attente respective. Lorsqu'il trouve un message à traiter, le processus adéquat pour ce genre d'application est déclenché, et lorsque le processus de traitement se termine, le processeur de lignes recommence à boucler dans le gestionnaire de processeur (LP\_DRIVER).

Dès lors, le dénominateur représente le temps passé par le processeur de lignes à vérifier l'existence de message(s) à traiter, plus, éventuellement, le temps passé à traiter ce(s) messages(s). Le total de ces deux temps doit donc être approximativement égal à quinze minutes.

Par contre, le numérateur représente uniquement le temps de traitement d'un (des) message(s), plus le temps passé à vérifier l'existence de ce(s) message(s) (Ce second temps est en fait, la proportion de temps effectivement consacré à traiter les messages de la ligne).

Une telle démarche est réalisée par la procédure DATATRIEVE 'LINE\_UTIL' en annexe page 143.

#### 4.1.2 Prise des mesures :

##### **- Agences et Front-ends :**

Pour une journée déterminée, nous avons d'abord extrait du fichier de comptabilité des appels 'ACNT.DAT' les enregistrements concernant ces deux types d'application (programme du type 'EXTRACT.MAR' du chapitre trois) et relatifs au processeur de lignes étudié. Nous avons ensuite déterminé l'heure la plus chargée pour un type d'application et pour le processeur de lignes étudié (grâce à une procédure du type 'INTERCERSEG' du chapitre trois). Enfin, au vu des résultats obtenus, nous avons sélectionné les enregistrements correspondants dans le fichier 'LINEPROC.SRT' et nous avons calculé le pourcentage d'utilisation du processeur de lignes, comme nous l'avons expliqué au point 4.1.1.

##### **- Bureautique :**

La méthode est la même que pour les types d'application précédents, avec toutefois la nécessité d'éviter un éventuel recoupement de la charge avec celle fournie par les Front-ends.

##### exemple :

(cfr figure 3.1 page 103). A 23 heures, il existe une charge non négligeable fournie par les Front-ends, qu'il ne faut pas cumuler avec la charge fournie par la bureautique. Les mesures concernant la bureautique ont donc dû être effectuées en dehors de la période de charge engendrée par les Front-ends.



## 4.1.3 Résultats obtenus (du 9 au 10 janvier 1986) :

* * * * *																			
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *										* * * * *									
* * * * *																			

















## 4.2 DISPONIBILITE DE LA MEMOIRE DES PROCESSEURS DE LIGNES.

### 4.2.1 Mesure de cette variable :

Pour mesurer cette variable, nous avons utilisé le fichier 'LINEPROC.SRT' décrit à la page 40. Grâce à ce fichier, il nous est possible de connaître le pourcentage de disponibilité de la mémoire d'un processeur de lignes déterminé durant une période statistique de plus ou moins quinze minutes. Il suffit de suivre la démarche suivante :

- Repérer dans le fichier l'enregistrement dont les numéros du noeud et du processeur sont identiques à ceux du processeur étudié (OFFICE\_CODE et LP\_NUMBER)
- et qui correspond à la période analysée (TIME).
- Enfin, appliquer la formule suivante afin d'en déduire le taux de disponibilité demandé :

$$\frac{\text{LP\_AVAIL\_FREE\_QUEUE}}{\text{LP\_LOAD\_FREE\_QUEUE}} \times 100$$

où LP-AVAIL-FREE-QUEUE représente le nombre total de blocs disponibles observés à chaque échantillon d'une minute

et LP-LOAD-FREE-QUEUE représente le nombre total de blocs libres mis à la disposition du processeur de lignes lors de l'initialisation.

Une telle démarche est réalisée par la procédure DATATRIEVE 'MEM\_UTIL' en annexe page 144.

### 4.2.2 Prise des mesures :

#### **- Agences et Front-ends :**

Pour une journée déterminée, nous avons d'abord extrait du fichier de comptabilité des appels 'ACNT.DAT' les enregistrements concernant ces deux types d'application (programme du type 'EXTRACT.MAR' du chapitre trois) et relatifs au processeur de lignes étudié. Nous avons ensuite déterminé l'heure la plus chargée pour un type d'application et pour le processeur de lignes étudié (grâce à une procédure du type 'INTERCERSEG' du chapitre trois). Enfin, au vu des résultats du point précédent, nous avons sélectionné les enregistrements correspondants dans le fichier 'LINEPROC.SRT' et nous avons calculé le pourcentage de disponibilité de la mémoire du processeur de lignes, comme nous l'avons expliqué au point 4.2.1.



- Bureautique :

La méthode est la même que pour les types d'application précédents, avec toutefois la nécessité d'éviter un éventuel recoupement de la charge avec celle fournie par les Front-ends.

exemple :

(cfr figure 3.1 page 103). A 23 heures, il existe une charge non négligeable fournie par les Front-ends, qu'il ne faut pas cumuler avec la charge fournie par la bureautique. Les mesures concernant la bureautique ont donc dû être effectuées en dehors de la période de charge engendrée par les Front-ends.





*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	
*****																	





### 4.3 POURCENTAGE D'UTILISATION DES TRONCS.

#### 4.3.1 Mesure de cette variable :

Pour mesurer cette variable, nous avons utilisé le fichier 'TRUNK.SRT' décrit à la page 42. Grâce à ce fichier, il nous est possible de connaître le pourcentage d'utilisation d'un tronc déterminé, pour une période statistique de plus ou moins quinze minutes. Il suffit de suivre la démarche suivante :

- Repérer dans le fichier, l'enregistrement dont les numéros des noeuds 'local' et 'remote' correspondent à ceux du tronc étudié (OFFICE\_CODE, LOCAL\_TRUNK\_NO, REMOTE\_NODE\_NO et REMOTE\_TRUNK\_NO)
- et qui correspond à la période analysée (TIME).
- Enfin, appliquer la formule suivante afin d'en déduire le pourcentage d'utilisation demandé :

$$\frac{(\text{FRAMES\_RECVD} - (\text{IDLE\_SLOTS} * 256 / 40))}{\text{FRAMES\_RECVD}} * 100$$

Cette formule s'explique de la manière suivante :

Rappelons qu'un tronc envoie continuellement des trames même s'il n'y a pas de données à expédier. Dès lors, le pourcentage d'utilisation d'un tronc s'obtient en divisant le nombre de trames effectivement utilisées par le nombre de trames reçues par le noeud local.

Le nombre de trames effectivement utilisées par un tronc, est la différence entre le nombre total de trames reçues et le nombre de trames vides. Le nombre de trames vides est égal au nombre d'"idle-slots" multiplié par 256 (car dans l'enregistrement, ce nombre est 256 fois plus petit que la valeur réelle) divisé par 40 (car une trame se compose de 40 slots).

Une telle démarche est réalisée par la procédure DATATRIEVE 'TRUNK\_UTIL' en annexe page 145.

#### 4.3.2 Prise des mesures :

Nous avons d'abord dû déterminer l'ensemble des troncs utilisés par chaque type d'application aux heures les plus chargées; ceci s'effectuait grâce à la connaissance de l'algorithme de routage décrit à la page 49.

Ensuite, par type d'application, nous avons effectué les différentes mesures :

##### **- Agences et Front-ends :**

Comme pour les deux variables précédentes, nous avons, pour une journée déterminée, extrait du fichier de comptabilité des appels 'ACNT.DAT' les enregistrements concernant ces deux types d'application (programme du type 'EXTRACT.MAR' du chapitre trois) et ensuite, nous avons déterminé l'heure la plus chargée par type d'application (grâce à une procédure du type 'INTERCERSEG' du chapitre trois). Enfin, au vu des résultats obtenus, nous avons sélectionné les enregistrements correspondants dans le fichier 'TRUNK.SRT', et nous avons calculé le pourcentage d'utilisation des troncs, comme nous l'avons expliqué au point 4.3.1.

##### **- Bureautique :**

Même type de remarque que pour les deux variables précédentes.



### 4.3.3 Résultats obtenus (du 9 au 10 janvier 1986):

TAUXTR	NOEUDDES	NOEUDORI	AP	VIT Kbps	ROUT
32.31	81	80	1	64.0	1
32.95	81	80	1	64.0	1
32.10	81	80	1	64.0	1
33.03	81	80	1	64.0	1
13.24	83	80	1	64.0	1
7.47	83	80	1	64.0	1
19.06	80	82	1	64.0	1
23.27	80	82	1	64.0	1
39.46	11	83	1	19.2	1
43.33	11	83	1	19.2	1
50.35	31	83	1	9.6	1
69.62	31	83	1	9.6	1
19.33	13	83	1	64.0	1
15.14	13	83	1	64.0	1
10.77	80	84	1	64.0	1
6.73	80	84	1	64.0	1
58.82	30	84	1	9.6	1
43.18	30	84	1	9.6	1
42.96	12	84	1	64.0	1
39.62	12	84	1	64.0	1
68.59	32	84	1	9.6	1
20.80	32	84	1	9.6	1
17.65	80	85	1	64.0	1
14.61	80	85	1	64.0	1
82.83	34	85	1	9.6	1
66.88	34	85	1	9.6	1
38.24	23	86	1	9.6	1
67.33	23	86	1	9.6	1
36.49	24	81	1	64.0	1
22.73	24	81	1	64.0	1
27.37	81	86	1	64.0	1
26.81	81	86	1	64.0	1
5.64	81	86	1	64.0	1
5.06	81	86	1	64.0	1
2.56	85	24	1	9.6	0
2.54	85	24	1	9.6	0
2.54	24	85	1	9.6	0
2.52	24	85	1	9.6	0
1.40	82	84	1	9.6	0
1.46	82	84	1	9.6	0
1.39	84	82	1	9.6	0
1.46	84	82	1	9.6	0
2.55	13	86	1	9.6	0
2.53	13	86	1	9.6	0
2.53	86	13	1	9.6	0
2.51	86	13	1	9.6	0
2.54	84	85	1	9.6	0
1.59	84	85	1	9.6	0

TAUXTR	NOEUDDES	NOEUDORI	AP	VIT Kbps	ROUT
2.59	85	84	1	9.6	0
2.57	85	84	1	9.6	0
10.12	80	81	2	64.0	1
9.90	80	81	2	64.0	1
9.97	80	81	2	64.0	1
10.02	80	81	2	64.0	1
14.53	86	81	2	64.0	1
13.81	86	81	2	64.0	1
8.30	81	24	2	64.0	1
11.33	81	24	2	64.0	1
5.07	85	80	2	64.0	1
5.16	85	80	2	64.0	1
2.58	84	80	2	64.0	1
2.48	84	80	2	64.0	1
12.24	84	32	2	9.6	1
14.13	84	32	2	9.6	1
6.00	82	80	2	64.0	1
7.01	82	80	2	64.0	1
4.33	80	83	2	64.0	1
3.29	80	83	2	64.0	1
5.21	84	12	2	64.0	1
5.95	84	12	2	64.0	1
18.81	85	34	2	9.6	1
18.63	85	34	2	9.6	1
23.35	86	23	2	9.6	1
14.89	86	23	2	9.6	1
4.23	83	13	2	64.0	1
4.10	83	13	2	64.0	1
18.60	83	31	2	9.6	1
17.52	83	31	2	9.6	1
12.86	83	11	2	19.2	1
12.44	83	11	2	19.2	1
3.57	85	24	2	9.6	0
2.87	85	24	2	9.6	0
2.94	24	85	2	9.6	0
2.82	24	85	2	9.6	0
2.53	82	84	2	9.6	0
2.54	82	84	2	9.6	0
2.55	84	82	2	9.6	0
2.57	84	82	2	9.6	0
3.29	13	86	2	9.6	0
3.62	13	86	2	9.6	0
4.03	86	13	2	9.6	0
4.04	86	13	2	9.6	0
2.77	84	85	2	9.6	0
2.62	84	85	2	9.6	0
2.81	85	84	2	9.6	0
2.67	85	84	2	9.6	0
19.21	83	11	3	19.2	1
20.31	83	11	3	19.2	1
18.40	83	11	3	19.2	1



TAUXTR	NOEUDDES	NOEUDORI	AP	VIT Kbps	ROUT
6.08	80	83	3	64.0	1
6.02	80	83	3	64.0	1
10.25	80	81	3	64.0	1
10.02	80	81	3	64.0	1
10.44	80	81	3	64.0	1
10.30	80	81	3	64.0	1
7.67	80	84	3	64.0	1
6.66	80	84	3	64.0	1
7.46	84	12	3	64.0	1
6.34	84	12	3	64.0	1
8.78	81	24	3	64.0	1
8.24	81	24	3	64.0	1
11.69	80	81	3	64.0	1
10.44	80	81	3	64.0	1
12.07	80	81	3	64.0	1
10.25	80	81	3	64.0	1
6.28	80	82	3	64.0	1
4.84	80	82	3	64.0	1
4.15	80	82	3	64.0	1
4.27	80	82	3	64.0	1
8.43	80	85	3	64.0	1
5.93	80	85	3	64.0	1
13.37	80	85	3	64.0	1
10.94	80	85	3	64.0	1
10.89	80	85	3	64.0	1
12.06	80	85	3	64.0	1
10.68	80	83	3	64.0	1
10.60	80	83	3	64.0	1
2.53	85	24	3	9.6	0
2.52	85	24	3	9.6	0
2.55	24	85	3	9.6	0
2.54	24	85	3	9.6	0
1.93	82	84	3	9.6	0
1.42	82	84	3	9.6	0
2.50	84	82	3	9.6	0
2.53	84	82	3	9.6	0
2.54	13	86	3	9.6	0
2.52	13	86	3	9.6	0
2.54	86	13	3	9.6	0
2.52	86	13	3	9.6	0
2.53	84	85	3	9.6	0
2.58	84	85	3	9.6	0
2.60	85	84	3	9.6	0
2.57	85	84	3	9.6	0

LEGENDE : Type d'application : Front-ends = valeur 1  
 Agences = valeur 2  
 bureautique = valeur 3.  
 ROUT = 1 si le tronc est utilisé par l'algorithme  
 = 0 sinon.



#### 4.4 TEMPS D'EXISTENCE D'UN 'CALL'.

##### Remarque :

Dans un premier temps, nous avons voulu avoir un aperçu de la distribution de la durée des 'calls' pour les Agences et les Front-ends. Pour cela, nous avons extrait les enregistrements correspondant à chaque type d'application du fichier de comptabilité des appels 'ACNT.DAT', ce qui nous fournissait deux fichiers 'AGENCES.DAT' et 'FEP.DAT' dont les structures sont les mêmes que celles du fichier de comptabilité des appels 'ACNT.DAT'. Sur ces deux fichiers, nous avons exécuté deux procédures DATATRIEVE du type de celle se trouvant en annexe page 146. Nous avons donc obtenu deux fichiers en annexe pages 147 et suivantes, avec lesquels nous avons réalisé les graphiques aux pages 128 et 129. Ces deux graphiques confirment en fait la durée moyenne de plus ou moins soixante secondes que nous avons observée pour chaque type d'application.

##### 4.4.1 Mesure de cette variable :

La variable est directement observable dans la zone 'CALL\_DURATION' des enregistrements du fichier de comptabilité des appels 'ACNT.DAT'.

##### 4.4.2 Prise des mesures :

###### **- Agences et Front-ends :**

Pour une journée déterminée, nous avons d'abord extrait du fichier de comptabilité des appels 'ACNT.DAT' les enregistrements correspondant à chaque type d'application. Ensuite, nous avons déterminé l'heure la plus chargée pour un type d'application (grâce à une procédure du type 'INTERCERSEG' du chapitre trois). Enfin, grâce au programme 'DATE.MAR' (chapitre trois), nous avons retenu les enregistrements durant l'heure la plus chargée pour chaque type d'application.

###### **- Bureautique :**

La méthode est la même que pour les deux types d'application précédents avec toutefois la nécessité de regrouper deux fichiers consécutifs à cause de la durée assez longue des 'calls'.

# *DISTRIBUTION DUREE DES CALLS (AGENCES)*

20/21 NOVEMBRE 1985

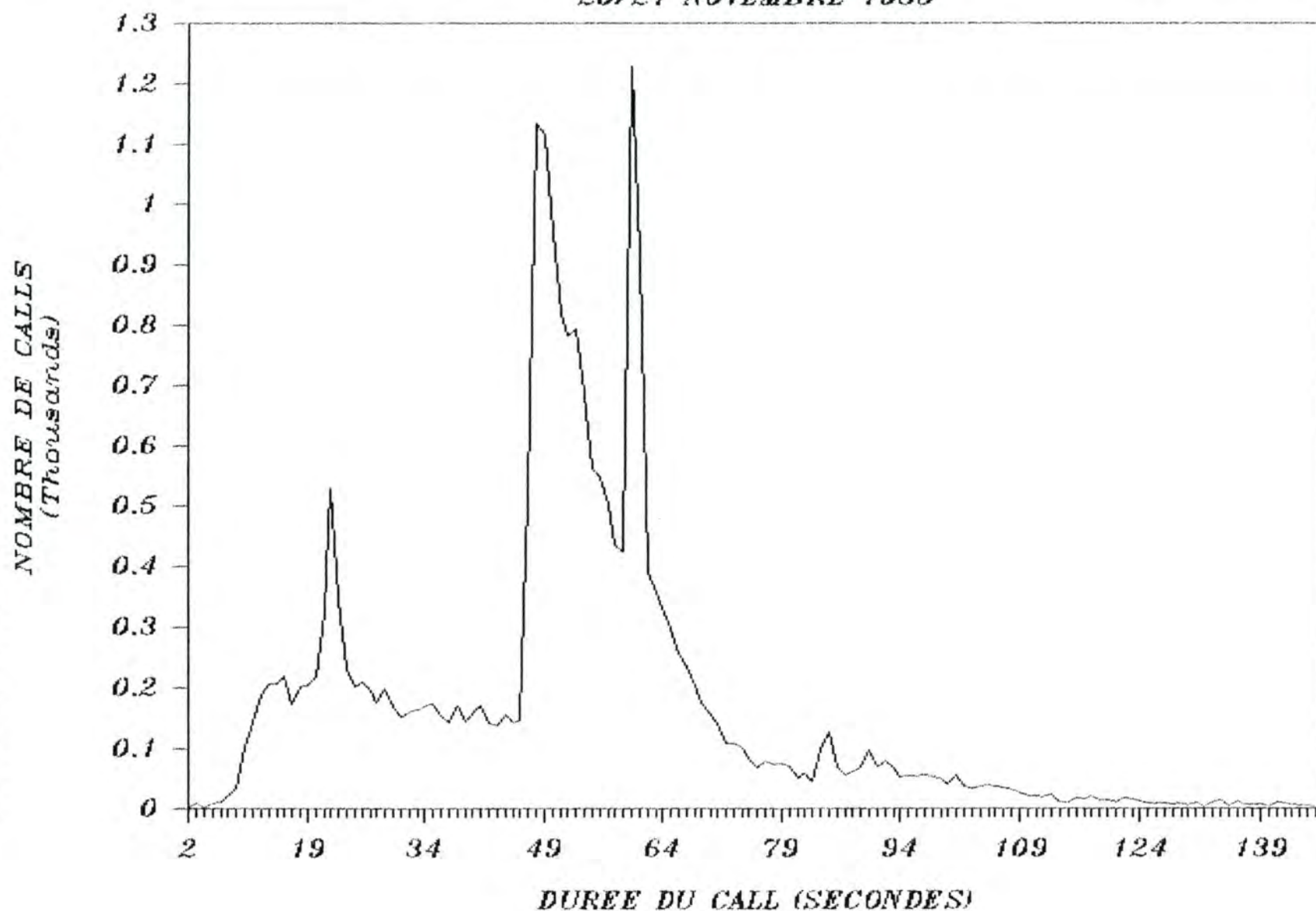


Figure 4.1 : Distribution des 'calls' (Agences).

# *DISTRIBUTION DUREE CALLS (FRONT-ENDS)*

21/22 NOVEMBRE 1985

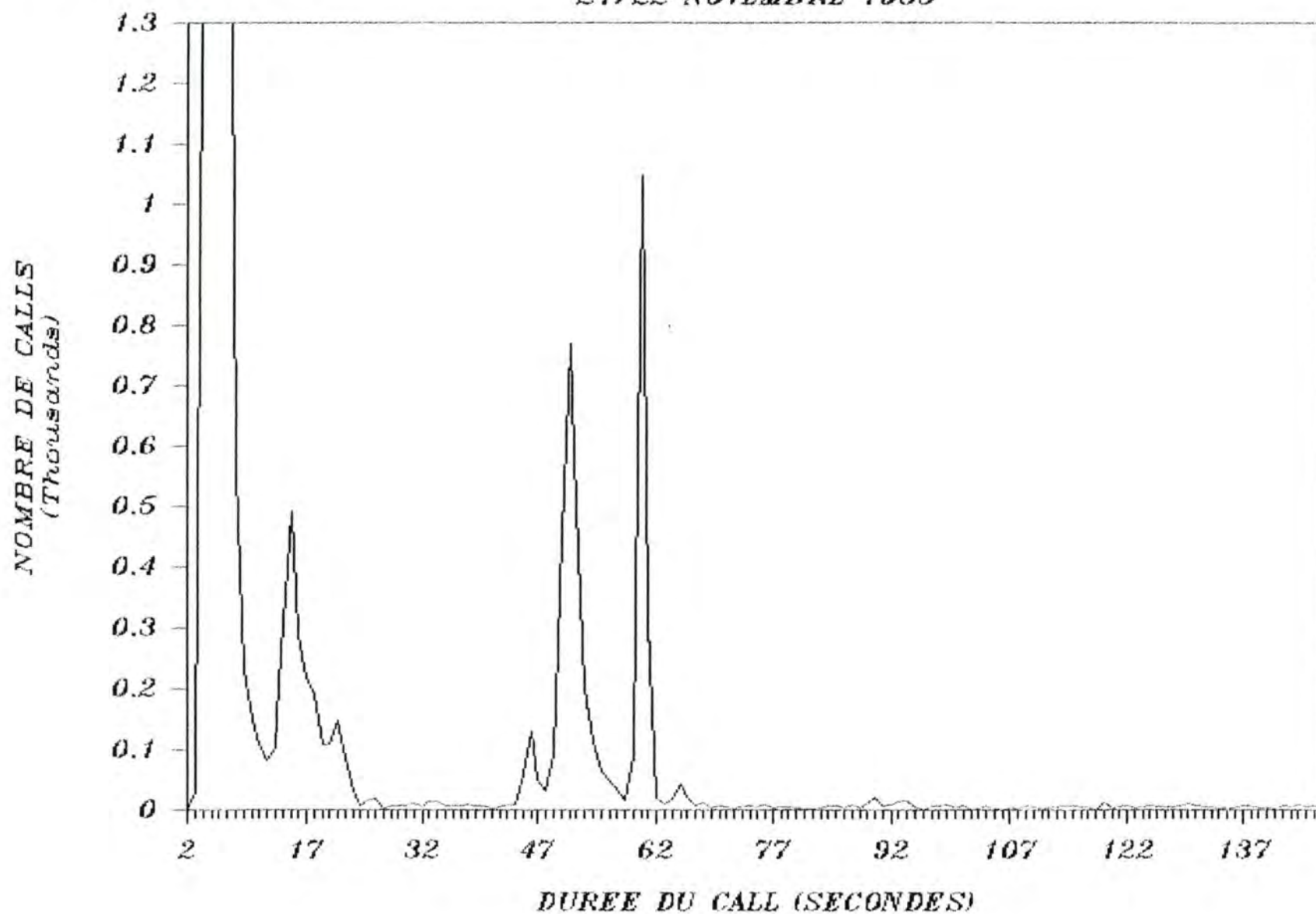


Figure 4.2 : Distribution des 'calls' (Front-ends).



#### 4.4.3 Résultats obtenus (du 9 au 10 janvier 1986):

* * * * *	* * * * *	* * * * *	* * * * *
* TEMPS	* AP	* SEG	*
* secondes	*	*	*
* * * * *	* * * * *	* * * * *	*
* 614	* 1	* 2353	*
* 60	* 1	* 3	*
* 500	* 1	* 1676	*
* 47	* 1	* 11	*
* 1300	* 1	* 4343	*
* 6	* 1	* 10	*
* 499	* 1	* 1626	*
* 60	* 1	* 3	*
* 432	* 1	* 1253	*
* 7	* 1	* 9	*
* 48	* 1	* 11	*
* 159	* 1	* 593	*
* 15	* 1	* 42	*
* 169	* 1	* 626	*
* 19	* 1	* 54	*
* 211	* 1	* 809	*
* 308	* 1	* 1107	*
* 111	* 2	* 114	*
* 100	* 2	* 44	*
* 48	* 2	* 10	*
* 34	* 2	* 75	*
* 51	* 2	* 24	*
* 102	* 2	* 262	*
* 51	* 2	* 26	*
* 90	* 2	* 21	*
* 49	* 2	* 14	*
* 84	* 2	* 251	*
* 50	* 2	* 22	*
* 42	* 2	* 43	*
* 89	* 2	* 191	*
* 56	* 2	* 41	*
* 58	* 2	* 54	*
* 56	* 2	* 46	*
* 56	* 2	* 44	*
* 58	* 2	* 33	*
* 84	* 2	* 175	*
* 90	* 2	* 21	*
* 53	* 2	* 33	*
* 58	* 2	* 50	*
* 49	* 2	* 17	*
* 26	* 2	* 54	*
* 68	* 2	* 27	*
* 12	* 2	* 33	*
* 51	* 2	* 24	*
* * * * *	* * * * *	* * * * *	*



#### 4.5 NOMBRE DE CIRCUITS VIRTUELS OUVERTS.

##### 4.5.1 Mesure de cette variable :

Pour mesurer cette variable, nous avons utilisé le fichier de comptabilité des appels 'ACNT.DAT' décrit à la page 36. Grâce à ce fichier, il nous est possible de connaître le nombre de circuits virtuels ouverts à un instant donné pour un processeur de lignes déterminé. En effet, il suffit de suivre la démarche suivante :

- Constituer un fichier avec toutes les adresses réseau (DNA) des utilisateurs reliés au processeur de lignes étudié.
- Sélectionner dans le fichier de comptabilité des appels 'ACNT.DAT', les enregistrements ayant une adresse réseau locale (LOCAL\_DNA) reprise dans le fichier constitué à l'étape précédente.
- Comptabiliser le nombre d'enregistrements pour lesquels l'intervalle de temps déterminé par l'heure de début (STIME) et de génération (TIME) de l'appel comprend l'heure demandée.

Une telle démarche est réalisée par la procédure DATATRIEVE 'FEP\_OPEN' en annexe page 152.

##### 4.5.2 Prise des mesures :

###### **- Agences et Front-ends :**

Selon le type d'application envisagé, nous avons d'abord extrait du fichier de comptabilité des appels 'ACNT.DAT' les différents enregistrements concernant ces deux types d'application au moyen de la procédure 'EXTRACT.MAR' du chapitre trois. Nous avons ensuite utilisé une procédure du type 'INTERCERSEG' (chapitre trois) afin de découvrir l'heure la plus chargée. Après avoir déterminé cette heure, nous avons exécuté la procédure 'DATE.MAR' et employé son fichier de sortie afin de suivre les deux derniers points de la démarche décrite au point 4.5.1.

###### **- Bureautique :**

La méthode appliquée est la même que celle utilisée pour les deux types d'application ci-dessus.



## 4.5.3 Résultats obtenus (du 9 au 10 janvier 1986) :

CIRCUIT	AP	BU	AL	F	B	MOY
21	1	266	35	1	0	1
20	1	266	35	1	0	1
31	1	1020	60	1	0	2
19	1	1020	60	1	0	2
22	1	1014	60	1	0	3
21	1	1014	60	1	0	3
20	1	1308	24	1	0	4
21	1	1308	24	1	0	4
22	1	290	53	1	0	5
23	1	290	53	1	0	5
20	1	1185	38	1	1	6
26	1	1185	38	1	1	6
4	2	284	30	0	3	7
2	2	284	30	0	3	7
5	2	266	35	1	0	8
2	2	266	35	1	0	8
1	2	1177	1	0	8	9
0	2	1177	1	0	8	9
4	2	391	42	0	1	10
3	2	391	42	0	1	10
6	2	341	43	0	1	11
5	2	341	43	0	1	11
4	2	1199	28	0	0	12
3	2	1199	28	0	0	12
4	2	345	49	0	1	13
8	2	345	49	0	1	13
2	2	563	21	0	0	14
0	2	563	21	0	0	14
4	2	341	43	0	0	15
2	2	341	43	0	0	15
4	2	1098	47	0	1	16
6	2	1098	47	0	1	16
1	2	402	36	0	0	17
0	2	402	36	0	0	17
0	2	469	36	0	0	18
1	2	469	36	0	0	18
2	2	488	25	0	0	19
3	2	488	25	0	0	19
1	2	445	35	0	1	20
1	2	445	35	0	1	20
2	2	271	38	0	0	21
2	2	271	38	0	0	21
6	2	1021	55	0	0	22
7	2	1021	55	0	0	22
4	2	1014	52	0	5	23
4	2	1014	52	0	5	23
4	2	1349	12	0	0	24
3	2	1349	12	0	0	24

CIRCUIT	AP	BU	AL	F	B	MOY
9	2	330	39	0	1	25
4	2	330	39	0	1	25
10	2	1020	60	1	0	26
5	2	1020	60	1	0	26
14	2	1051	57	0	1	27
5	2	1051	57	0	1	27
5	2	276	42	0	1	28
6	2	276	42	0	1	28
4	2	1014	60	1	0	29
2	2	1014	60	1	0	29
3	2	499	22	0	0	30
2	2	499	22	0	0	30
2	2	1308	24	1	0	31
1	2	1308	24	1	0	31
9	2	290	53	1	0	32
0	2	290	53	1	0	32
2	2	1014	60	0	0	33
12	2	1014	60	0	0	33
1	2	1417	4	0	3	34
0	2	1417	4	0	3	34
2	2	345	36	0	0	35
3	2	345	36	0	0	35
4	2	1185	38	1	1	36
2	2	1185	38	1	1	36
1	3	1098	47	0	1	37
2	3	1098	47	0	1	37
1	3	1177	1	0	8	38
1	3	1177	1	0	8	38
0	3	1014	52	0	5	39
0	3	1014	52	0	5	39
1	3	1417	4	0	3	40
2	3	1417	4	0	3	40
1	3	391	42	0	1	41
1	3	391	42	0	1	41
0	3	341	43	0	1	42
0	3	341	43	0	1	42
0	3	284	30	0	3	43
0	3	284	30	0	3	43
1	3	330	39	0	1	44
2	3	330	39	0	1	44
0	3	276	42	0	1	45
0	3	276	42	0	1	45

LEGENDE : Type d'application : Front-ends = valeur 1  
Agences = valeur 2  
bureautique = valeur 3.



## 4.6 POURCENTAGE DE SEGMENTS LOCAUX.

### 4.6.1 Mesure de cette variable :

Pour mesurer cette variable, nous avons utilisé le fichier de comptabilité des appels 'ACNT.DAT' décrit à la page 36. Grâce à ce fichier, il nous est possible de connaître le pourcentage de segments locaux à un noeud durant une période de charge donnée. En effet, il suffit de suivre la démarche suivante :

- Sélectionner dans le fichier de comptabilité des appels 'ACNT.DAT', les enregistrements ayant comme numéro de noeud local (OFFICE\_CODE) et de noeud éloigné (deux premiers caractères du REMOTE\_DNA) celui du noeud étudié.
- Comptabiliser pour les enregistrements retenus à l'étape précédente, le nombre de segments locaux envoyés et reçus durant la période de temps demandée (ceci est possible en analysant l'heure de début (STIME) et de génération (TIME) de l'appel).
- Calculer le pourcentage de segments locaux par rapport au nombre total de segments transitant durant la période étudiée.

Une telle démarche est réalisée par la procédure DATATRIEVE 'SEG\_LOC\_FEP' en annexe page 154.

### 4.6.2 Prise des mesures :

#### **- Agences et Front-ends :**

Selon le type d'application envisagé, nous avons d'abord extrait du fichier de comptabilité des appels 'ACNT.DAT' les différents enregistrements concernant ces deux types (Agences et Front-ends) au moyen de la procédure 'EXTRACT.MAR' du chapitre trois. Nous avons ensuite utilisé une procédure du type 'INTERCERSEG' (chapitre trois) afin de découvrir l'heure la plus chargée. Après avoir déterminé cette heure, nous avons exécuté la procédure 'DATE.MAR' (chapitre trois) et employé son fichier de sortie afin de suivre les points de la démarche décrite au point 4.6.1.

#### **- Bureautique :**

Nous ne fournissons pas de résultat en ce qui concerne la bureautique pour la raison suivante :

Toutes les mesures effectuées en ce qui concerne ce type d'application nous ont donné comme résultat 0 % de segments locaux à tous les noeuds supportant des lignes utilisées par la bureautique (Ceci est essentiellement dû à la fonction de transfert de fichiers réalisée avec une machine d'archivage reliée au noeud 80).



## 4.6.3 Résultats obtenus (du 13 au 23 novembre 1985):

TAUXSEGLOC	NI	AP	FEN	AFN	AFPN	AAF	AGENMF	MOY
25.49	100	1	1	31	120	35	1	1
25.11	100	1	1	31	120	35	1	1
22.18	100	1	1	31	120	35	1	1
64.85	100	1	1	144	52	12	0	2
67.92	100	1	1	144	52	12	0	2
67.68	100	1	1	144	52	12	0	2
37.08	100	1	1	83	126	19	0	3
43.22	100	1	1	83	126	19	0	3
37.04	100	1	1	83	126	19	0	3
15.22	100	1	1	46	152	0	0	4
15.53	100	1	1	46	152	0	0	4
13.17	100	1	1	46	152	0	0	4
62.21	100	1	1	117	67	0	0	5
60.66	100	1	1	117	67	0	0	5
69.13	100	1	1	117	67	0	0	5
51.25	100	1	1	74	111	0	0	6
40.57	100	1	1	74	111	0	0	6
42.95	100	1	1	74	111	0	0	6
60.20	100	2	1	31	120	35	1	7
61.16	100	2	1	31	120	35	1	7
61.00	100	2	1	31	120	35	1	7
1.94	100	2	0	0	0	90	1	8
1.60	100	2	0	0	0	90	1	8
2.38	100	2	0	0	0	90	1	8
2.57	100	2	0	0	0	157	1	9
2.58	100	2	0	0	0	157	1	9
2.98	100	2	0	0	0	157	1	9
92.60	100	2	1	144	52	12	0	10
92.53	100	2	1	144	52	12	0	10
88.79	100	2	1	144	52	12	0	10
79.52	100	2	1	83	126	19	0	11
83.08	100	2	1	83	126	19	0	11
78.90	100	2	1	83	126	19	0	11
99.45	100	2	1	46	152	0	0	12
99.22	100	2	1	46	152	0	0	12
94.38	100	2	1	46	152	0	0	12
99.00	100	2	1	117	67	0	0	13
99.43	100	2	1	117	67	0	0	13
93.24	100	2	1	117	67	0	0	13
98.21	100	2	1	74	111	0	0	14
98.70	100	2	1	74	111	0	0	14
98.17	100	2	1	74	111	0	0	14
0.01	100	2	0	0	0	42	0	15
0.02	100	2	0	0	0	42	0	15

* * * * *													
* * * * *													
* TAUXSEGLOC * NI * AP * FEN * AFN * AFPN * AAF * AGENMF * MOY *													
* * * * *													
* * * * *													
* 0.01 * 100 * 2 * 0 * 0 * 0 * 71 * 0 * 16 *													
* 0.02 * 100 * 2 * 0 * 0 * 0 * 71 * 0 * 16 *													
* 0.01 * 100 * 2 * 0 * 0 * 0 * 49 * 0 * 17 *													
* 0.02 * 100 * 2 * 0 * 0 * 0 * 49 * 0 * 17 *													
* 0.01 * 100 * 2 * 0 * 0 * 0 * 21 * 0 * 18 *													
* 0.02 * 100 * 2 * 0 * 0 * 0 * 21 * 0 * 18 *													
* 0.01 * 100 * 2 * 0 * 0 * 0 * 36 * 0 * 19 *													
* 0.02 * 100 * 2 * 0 * 0 * 0 * 36 * 0 * 19 *													
* 0.01 * 100 * 2 * 0 * 0 * 0 * 36 * 0 * 20 *													
* 0.02 * 100 * 2 * 0 * 0 * 0 * 36 * 0 * 20 *													
* 0.01 * 100 * 2 * 0 * 0 * 0 * 25 * 0 * 21 *													
* 0.02 * 100 * 2 * 0 * 0 * 0 * 25 * 0 * 21 *													
* 0.01 * 100 * 2 * 0 * 0 * 0 * 35 * 0 * 22 *													
* 0.02 * 100 * 2 * 0 * 0 * 0 * 35 * 0 * 22 *													
* * * * *													
* * * * *													

LEGENDE : Type d'application (AP) : Front-ends = valeur 1  
Agences = valeur 2.  
AGENMF : 1 = présence d'agences "fille" "mère"  
0 = absence d'agences "fille" "mère"



#### 4.7 DISPONIBILITE DU RESEAU.

##### 4.7.1 Mesure de cette variable :

Pour mesurer cette variable, nous avons utilisé le fichier de comptabilité des appels 'ACNT.DAT' décrit à la page 36. Grâce à ce fichier, il nous est possible de connaître le pourcentage d'appels acceptés et refusés par noeud durant une période de charge donnée. En effet, il suffit de suivre la démarche suivante :

- Sélectionner dans le fichier de comptabilité des appels 'ACNT.DAT', les enregistrements ayant un numéro de noeud local (OFFICE\_CODE) identique à celui étudié et dont la période de l'appel est entièrement comprise dans la période analysée.
- Analyser la zone "CLEAR\_CODE" des enregistrements retenus.
- Totaliser le nombre d'enregistrements correspondant à un appel accepté.
- Totaliser le nombre d'enregistrements correspondant à un appel refusé.
- Calculer le pourcentage d'appels acceptés et refusés.

Une telle démarche est réalisée par la procédure DATATRIEVE 'CLEAR\_CODE\_FEP' en annexe page 155.

##### 4.7.2 Prise des mesures :

###### **- Agences et Front-ends :**

Selon le type d'application envisagé, nous avons d'abord extrait du fichier de comptabilité des appels 'ACNT.DAT' les différents enregistrements concernant ces deux types (Agences et Front-ends) au moyen de la procédure 'EXTRACT.MAR' du chapitre trois. Nous avons ensuite utilisé une procédure du type 'INTERCERSEG' (chapitre trois), afin de découvrir l'heure la plus chargée. Après avoir déterminé cette heure, nous avons exécuté la procédure 'DATE.MAR' (chapitre trois) et employé son fichier de sortie afin de suivre les points de la démarche décrite au point 4.7.1.

###### **- Bureautique :**

Nous ne fournissons pas de résultat en ce qui concerne la bureautique parce que le nombre de 'calls' est trop petit pour pouvoir en tirer un enseignement correct. Ceci dit, les mesures effectuées pour ce type d'application ne nous ont jamais permis d'observer une disponibilité du réseau inférieure à 75 %.



## 4.7.3 Résultats obtenus (du 13 au 23 novembre 1985):

TAUXDIS											
BO	*	B1	AP	FEN	AFN	AFPN	AAF	MOY			
24.96	*	46.10	1	1	31	120	35	1			
23.09	*	48.02	1	1	31	120	35	1			
21.61	*	48.36	1	1	31	120	35	1			
26.49	*	53.31	1	1	31	120	35	1			
23.22	*	52.19	1	1	31	120	35	1			
15.91	*	80.54	1	1	144	52	12	2			
9.90	*	75.67	1	1	144	52	12	2			
10.76	*	84.63	1	1	144	52	12	2			
13.87	*	85.86	1	1	144	52	12	2			
13.61	*	84.29	1	1	144	52	12	2			
10.00	*	74.68	1	1	83	126	19	3			
10.19	*	84.14	1	1	83	126	19	3			
9.66	*	74.68	1	1	83	126	19	3			
5.21	*	77.39	1	1	83	126	19	3			
9.61	*	90.27	1	1	83	126	19	3			
7.01	*	65.85	1	1	46	152	0	4			
6.52	*	70.73	1	1	46	152	0	4			
3.47	*	72.62	1	1	46	152	0	4			
3.33	*	81.30	1	1	46	152	0	4			
0.09	*	85.25	1	1	46	152	0	4			
14.65	*	72.76	1	1	117	67	0	5			
7.89	*	43.37	1	1	117	67	0	5			
10.24	*	71.00	1	1	117	67	0	5			
9.25	*	72.32	1	1	117	67	0	5			
9.46	*	68.65	1	1	117	67	0	5			
7.66	*	71.79	1	1	74	111	0	6			
8.18	*	80.37	1	1	74	111	0	6			
13.69	*	81.94	1	1	74	111	0	6			
11.27	*	83.77	1	1	74	111	0	6			
10.39	*	84.37	1	1	74	111	0	6			
37.58	*	60.97	2	1	31	120	35	7			
37.19	*	61.34	2	1	31	120	35	7			
43.90	*	56.09	2	1	31	120	35	7			
36.85	*	63.14	2	1	31	120	35	7			

TAUXDIS											
BO	+	B1	AP	FEN	AFN	AFPN	AAF	MOY			
45.22	*	54.77	2	0	0	0	42	8			
44.52	*	55.47	2	0	0	0	42	8			
44.82	*	55.17	2	0	0	0	42	8			
2.56	*	97.43	2	0	0	0	42	8			
35.41	*	64.08	2	0	0	0	71	9			
15.93	*	68.85	2	0	0	0	71	9			
28.66	*	71.33	2	0	0	0	71	9			
5.52	*	94.33	2	0	0	0	71	9			
46.06	*	53.94	2	0	0	0	49	10			
44.26	*	55.73	2	0	0	0	49	10			
44.45	*	53.54	2	0	0	0	49	10			
2.65	*	97.34	2	0	0	0	49	10			
22.02	*	77.97	2	0	0	0	21	11			
47.40	*	52.59	2	0	0	0	21	11			
47.81	*	52.18	2	0	0	0	21	11			
0.00	*	100.00	2	0	0	0	21	11			
34.09	*	65.91	2	0	0	0	90	12			
33.25	*	66.74	2	0	0	0	90	12			
28.87	*	71.03	2	0	0	0	90	12			
2.88	*	97.11	2	0	0	0	90	12			
37.20	*	62.79	2	0	0	0	36	13			
16.33	*	65.19	2	0	0	0	36	13			
29.20	*	70.79	2	0	0	0	36	13			
4.54	*	95.45	2	0	0	0	36	13			
41.52	*	58.47	2	0	0	0	36	14			
45.89	*	54.10	2	0	0	0	36	14			
42.71	*	57.28	2	0	0	0	36	14			
2.98	*	97.01	2	0	0	0	36	14			
34.82	*	64.78	2	0	0	0	25	15			
17.14	*	63.31	2	0	0	0	25	15			
28.15	*	71.62	2	0	0	0	25	15			
6.32	*	93.67	2	0	0	0	25	15			
46.87	*	52.86	2	0	0	0	35	16			
33.79	*	66.01	2	0	0	0	35	16			
43.38	*	56.61	2	0	0	0	35	16			
0.00	*	100.00	2	0	0	0	35	16			
35.82	*	63.94	2	0	0	0	157	17			
43.78	*	56.21	2	0	0	0	157	17			
38.81	*	60.56	2	0	0	0	157	17			
3.78	*	96.21	2	0	0	0	157	17			





## CHAPITRE 5

### METHODE D'EXPLOITATION STATISTIQUE DES RESULTATS.

-----

#### Résumé :

L'objectif de ce chapitre est dans un premier temps, de présenter la méthode statistique utilisée pour exploiter les résultats obtenus au chapitre précédent. Dans un deuxième temps, nous présentons les différents résultats de modélisation que nous avons obtenus pour chaque variable à expliquer.

-----

## 5.1 INTRODUCTION.

Afin de mettre en évidence la dépendance des variables à expliquer  $Y_i$  en fonction des variables explicatives  $X_i$  (connues à l'avance), nous avons utilisé des méthodes de type 'modèle linéaire normal'.

Rappelons qu'un modèle linéaire normal suppose que :

$$Y_i = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p + \epsilon_i$$

où les  $\epsilon_i$  sont des erreurs normales de moyenne nulle et d'écart-type  $\sigma$ , constant d'une observation à l'autre, les  $\epsilon_i$  étant indépendantes les unes des autres.

Les  $X_i$  sont des variables, en principe quantitatives, mais la méthode peut être étendue à des variables qualitatives en éclatant chacune d'elles en autant de variables que la variable initiale comporte de modalités moins une.

Comme la plupart des variables que nous devons expliquer sont des taux d'utilisation (compris entre 0 et 1), la méthode traditionnelle ne peut être directement appliquée car la distribution n'est pas normale et la variance des erreurs n'est pas constante. Nous devons dès lors choisir entre deux solutions :

- a) ou bien effectuer une transformation du type  $\ln [Y / (1 - Y)]$  qui transforme la variable initiale en une variable comprise entre  $-\infty$  et  $+\infty$ .
- b) ou bien envisager un modèle linéaire généralisé tel que, en posant  $\mu_i$  l'espérance mathématique de  $Y_i$ , une fonction de  $\mu_i$  est combinaison linéaire des  $X_j$ .

Par exemple :

$$\ln (- \ln \mu_i) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p \quad (5.1)$$

ou toute autre fonction de  $\mu_i$ . Cette fonction est appelée 'fonction de lien'.

Le modèle (5.1) est généralisé du point de vue de la distribution : les  $Y_i$ , au lieu d'avoir une distribution normale, sont supposés avoir une distribution binomiale, ou binomiale généralisée, ...

Dans notre étude, après divers essais, nous n'avons utilisé que les modèles linéaires normaux avec le premier type de transformation. Nous rappelons les principes essentiels de ces modèles au point suivant.

## 5.2 LES MODELES LINEAIRES NORMAUX. (cours de "statistique appliquée à la gestion" de Madame NOIRHOMME)

Nous développons ici les concepts fondamentaux des modèles linéaires normaux; le lecteur intéressé pourra toujours se référer au cours de Madame NOIRHOMME, mentionné ci-dessus.

### exemple introductif :

Nous pouvons très bien supposer qu'une variable physique  $y$  soit fonction linéaire d'une autre variable quantitative  $x$  que nous pouvons maîtriser.

exemple :  $y$  = allongement d'un corps en fonction de la température  $x$  que nous pouvons maîtriser.

Nous allons effectuer plusieurs observations de  $y$  en fixant  $x$  à différents 'niveaux'.

Si le modèle rectiligne est vrai, les observations  $Y_{i,k}$  ( $k^{\text{ème}}$  observation de  $y$  pour le  $i^{\text{ème}}$  niveau de  $x$ ) pourront se décomposer comme suit :

$$Y_{i,k} = \beta_0 + \beta_1 x_i + \epsilon_{i,k} \quad (5.2)$$

Les observations sont donc entachées d'une erreur aléatoire qui, si la mesure est bien faite, sont indépendantes et de moyenne nulle. Nous supposons en outre qu'elles ont une distribution normale de variance constante.

(5.2) peut aussi se décomposer comme suit :

$$\begin{array}{ccccccc} ! & Y_{1,1} & ! & ! & x_1 & ! & ! & \beta_0 & ! & ! & \epsilon_{1,1} & ! \\ ! & . & ! & ! & . & ! & ! & . & ! & ! & . & ! \\ ! & . & ! & ! & . & ! & ! & \beta_1 & ! & ! & . & ! \\ ! & Y_{1,n1} & ! & ! & . & ! & ! & . & ! & ! & \epsilon_{1,n1} & ! \\ ! & Y_{2,1} & ! & ! & . & ! & ! & x_2 & ! & ! & \epsilon_{2,1} & ! \\ ! & . & ! & ! & . & ! & ! & . & ! & ! & . & ! \\ ! & . & ! & ! & . & ! & ! & . & ! & ! & . & ! \\ ! & Y_{2,n2} & ! & = & ! & . & ! & x_2 & ! & + & ! & \epsilon_{2,n2} & ! \\ ! & . & ! & ! & . & ! & ! & . & ! & ! & . & ! \\ ! & . & ! & ! & . & ! & ! & . & ! & ! & . & ! \\ ! & . & ! & ! & . & ! & ! & . & ! & ! & . & ! \\ ! & Y_{p,1} & ! & ! & . & ! & ! & x_p & ! & ! & \epsilon_{p,1} & ! \\ ! & . & ! & ! & . & ! & ! & . & ! & ! & . & ! \\ ! & . & ! & ! & . & ! & ! & . & ! & ! & . & ! \\ ! & Y_{p,np} & ! & ! & 1 & ! & ! & x_p & ! & ! & \epsilon_{p,np} & ! \end{array}$$

ou encore

$$\epsilon Y = Ab \quad \text{où } b = \begin{array}{c} ! \beta_0 ! \\ ! \beta_1 ! \end{array}$$

Dans la pratique, pour un tel problème, on demande d'estimer les deux paramètres  $\beta_0$  et  $\beta_1$ , c'est à dire la droite.



DE MANIERE TRES GENERALE, tous les modèles linéaires peuvent s'exprimer de la façon suivante :

$$Y = Ab + E$$

avec  $Y$  un vecteur d'observations

$A$  une matrice de coefficients

$b$  un vecteur de paramètres

$E$  un vecteur d'erreurs normales, indépendantes de moyenne nulle et de variance  $\sigma^2$ .

La méthode permet

- d'estimer une combinaison linéaire des  $\beta_i$
- d'estimer les  $\beta_i$  conjointement
- d'estimer la variance  $\sigma^2$
- de comparer le modèle à un modèle plus complexe.

### Estimation des paramètres.

#### Estimation de $f^T b$ :

Nous prendrons comme estimateur  $\hat{b}$  de  $b$ , le vecteur qui minimise la longueur du vecteur d'erreur (estimée)  
 $\hat{e} = Y - A\hat{b}$ ;  $\hat{b}$  sera donc un estimateur au sens des moindres carrés.

La solution de l'estimation, quand  $A$  est de rang maximum est :

$$\hat{b} = (A^T A)^{-1} A^T Y$$

Lorsque l'estimation porte sur une combinaison linéaire des paramètres (notée  $f^T b$  où  $f$  est un vecteur de coefficients), le résultat de l'estimation est :

$$f^T \hat{b} = f^T (A^T A)^{-1} A^T Y.$$

On peut montrer que  $f^T \hat{b}$  est, parmi les estimateurs sans biais, celui qui a la plus petite variance. Cette variance est la suivante :

$$\text{var } f^T \hat{b} = \sigma^2 f^T (A^T A)^{-1} f$$

Nous savons que, par hypothèse, les différentes observations  $Y_i$  sont des variables normales indépendantes; donc  $f^T \hat{b}$  est une variable aléatoire normale de moyenne  $f^T b$  et de variance  $\sigma^2 f^T (A^T A)^{-1} f$ , puisqu'elle peut s'exprimer comme combinaison linéaire des  $Y_i$ . Nous pouvons dès lors obtenir soit un intervalle de confiance pour  $f^T b$ , soit une estimation simultanée de  $q$  combinaisons linéaires des paramètres, à la condition de connaître  $\sigma$ .

### Estimation conjointe des $\beta_i$ :

Pour estimer simultanément  $q$  combinaisons linéaires des paramètres, il faut déterminer une région de confiance de niveau d'incertitude  $\alpha$  connu.

On peut démontrer que :

$$\Pr \bigcap_{k=1}^q \{ f_k^T \underline{\hat{\beta}} - \sigma [ ]_k^{1/2} \leq f_k^T \underline{\beta} \leq f_k^T \underline{\hat{\beta}} + \sigma [ ]_k^{1/2} \} = 1 - \alpha$$

$$\text{avec } [ ]_k = s \cdot Q_F(1-\alpha; s, n-p) \cdot f_k^T \cdot (A^T A)^{-1} \cdot f_k$$

$$\text{où } \sigma^2 = \frac{\text{SCE}}{n-p}$$

-  $s$  est le nombre de vecteurs  $f_i^T$  linéairement indépendants.

### Estimation de $\sigma^2$ :

Nous pouvons également estimer la variance de l'erreur :

Pour cela, nous pouvons montrer que  $\text{SCT} = \text{SCN} + \text{SCE}$

où SCT = somme des carrés totale ( $\underline{Y}^T \underline{Y}$ ). Il s'agit du carré de la longueur du vecteur des observations.

SCN = somme des carrés normale ( $\|\underline{A}\underline{\hat{\beta}}\|^2$ ). Il s'agit du carré de la longueur de l'estimation.

SCE = somme des carrés des erreurs ( $\|\hat{\underline{e}}\|^2$ ). Il s'agit du carré de la longueur du vecteur d'erreur estimé. Cette somme est obtenue par différence entre SCT et SCN.

On peut montrer que  $(\text{SCE} / \sigma^2)$  a une distribution chi-carré à  $n-p$  degrés de liberté (où  $p$  = rang de la matrice  $A$  et  $n$  est le nombre d'observations).

Ceci nous permet d'obtenir directement un intervalle de confiance pour  $\sigma^2$  car nous connaissons sa distribution.

### Comparaison du modèle à un modèle plus complexe :

C'est ici qu'on introduit la notion de modèle emboîté dans un autre. Ceci nous permet, par exemple de tester la validité d'un modèle choisi a priori ( $M_1$ ), par rapport au modèle le plus général ( $M$ ) où l'on suppose seulement que les observations sont groupées en cellules ( $Y_{1,1}, \dots, Y_{i,n_i}$ ) de moyenne  $\mu_i$  (Pour cela, il est nécessaire qu'il y ait plusieurs observations par cellule).



La décision d'accepter le modèle  $M_1$  se fait sur base de la variable ' $\underline{r}$ ' suivante :

$$\underline{r} = \frac{SCV / (rangA - rangA_1)}{SCE / (n - rangA)} \quad \text{où } SCV = (SCE_1 - SCE)$$

SCE = somme des carrés  
des erreurs pour M.  
SCE<sub>1</sub> = somme des carrés  
des erreurs pour M<sub>1</sub>.

La variable ' $\underline{r}$ ' est une variable F de Snedecor à  $(rangA - rangA_1)$  et  $(n - rangA)$  degrés de liberté.

La région critique s'écrit :

$$R_C = \{w : \underline{r}(w) > Q_{F(p-q;n-p)}(1 - \alpha)\}$$

si  $\alpha$  est l'erreur de première espèce  
p est le rang de A  
q est le rang de A<sub>1</sub>

et l'hypothèse  $M_1$  sera rejetée par rapport à M, au niveau  $\alpha$  si l'observation a fourni un  $\underline{r}(w)$  supérieur au quantile  $Q_{F(p-q;n-p)}(1 - \alpha)$ .

On peut également conclure en utilisant une probabilité de dépassement liée à l'échantillon (probabilité d'avoir un échantillon plus défavorable). Dans ce cas, on obtient :

$$PD(w) = 1 - F_{F(p-q;n-p)}(\underline{r}(w)).$$



### 5.3 MODELES OBTENUS POUR CHAQUE VARIABLE DE PERFORMANCE.

#### 5.3.1 Hypothèses de départ :

Etant donné que nous avons utilisé les modèles linéaires normaux, il est nécessaire de vérifier les hypothèses de départ :

- Les erreurs sont normales et de moyenne nulle
- La variance des erreurs  $\sigma^2$  est constante d'une observation à l'autre.

Nous avons vérifié la validité de ces deux hypothèses pour des modèles utilisant un type particulier de transformation de la variable à expliquer.

#### \* Transformation du type $\ln (Y / (1 - Y))$ .

- - - - -

Les tests ont été effectués pour la variable à expliquer TAUXPL (taux d'utilisation des processeurs de lignes) et pour les données concernant le type d'application agences.

Nous présentons les résultats obtenus ci-après :

- Erreurs normales et de moyenne nulle (cfr figure 5.3 page 149).

Les erreurs observées sont reportées sur l'axe horizontal. L'axe vertical correspond aux valeurs normales attendues. Nous remarquons dans la figure 5.3 que les valeurs se rangent le long d'une droite indiquée par le symbole '//', ce qui signifie que l'hypothèse de normalité des données peut être raisonnablement acceptée.

- Variance des erreurs  $\sigma^2$  constante d'une observation à l'autre (cfr figure 5.4 page 150).

Ce graphique représentant les erreurs en fonction des valeurs transformées de la variable à expliquer nous permet d'admettre que la variance des erreurs est relativement stable d'une observation à l'autre (la dispersion des erreurs autour de leur moyenne n'est pas trop grande).

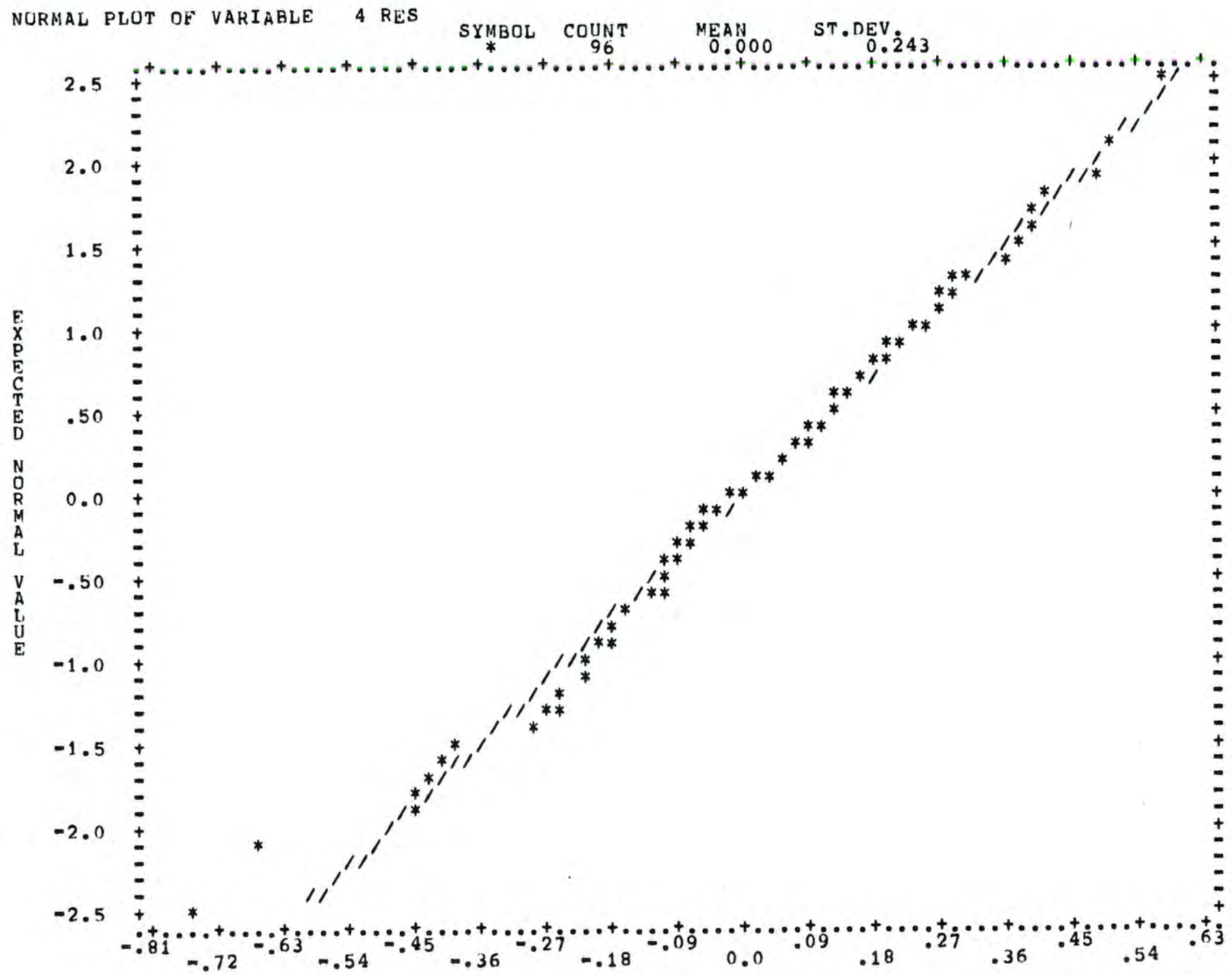
#### \* Transformation du type $\ln (Y)$ .

- - - - -

Les tests ont été effectués pour la variable à expliquer TAUXMEM (disponibilité de la mémoire des processeurs de lignes). L'analyse des graphiques obtenus (cfr figure 5.5 page 150 et figure 5.6 page 151) nous permet d'aboutir aux mêmes conclusions que pour le type de transformation précédent.

Par souci de complétude, il serait nécessaire d'effectuer les mêmes tests pour toutes les autres variables à étudier.

Figure 5.3 : Graphique normal des erreurs pour la variable de performance TAUXPL.



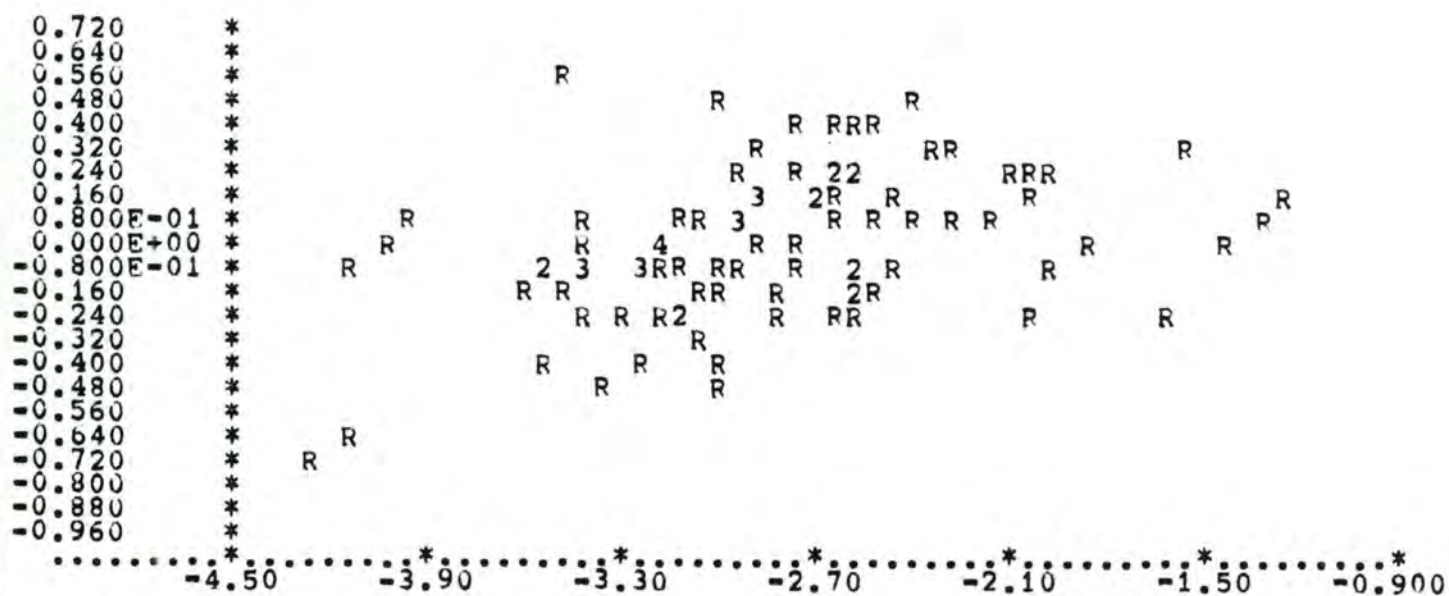


Figure 5.4 : Graphique des erreurs en fonction des valeurs transformées de la variable TAUXPL.

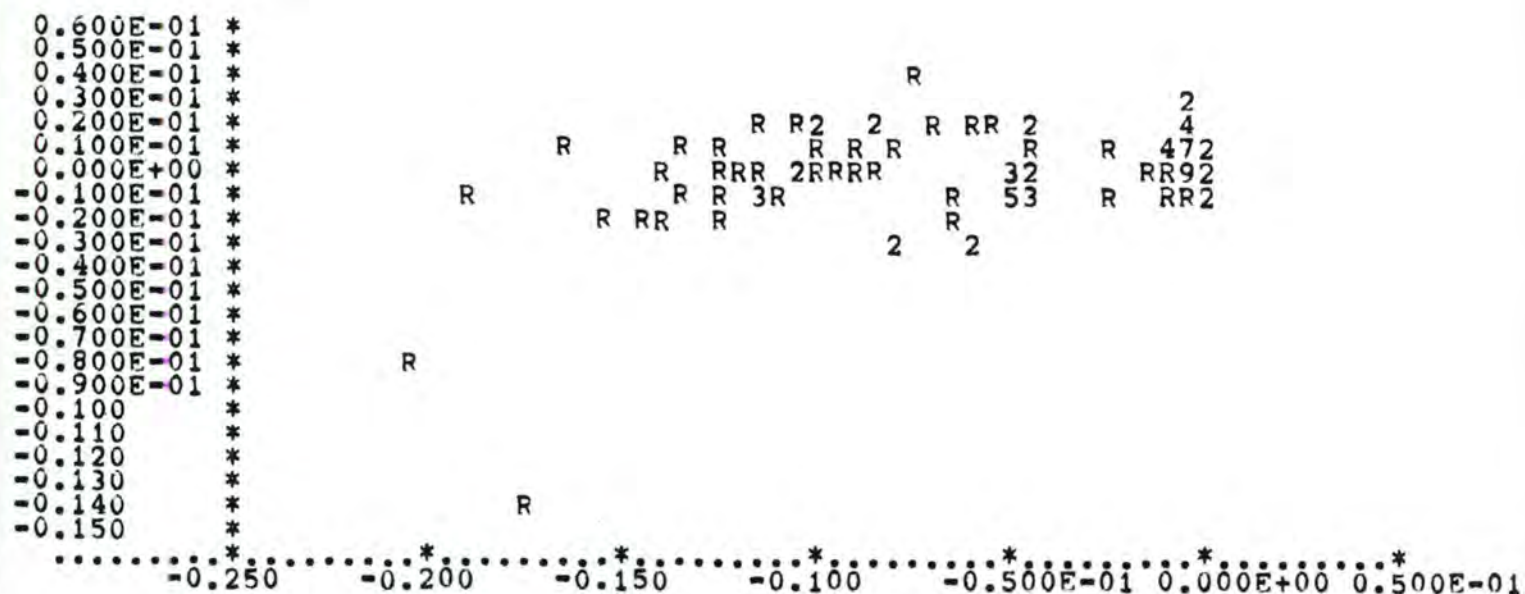
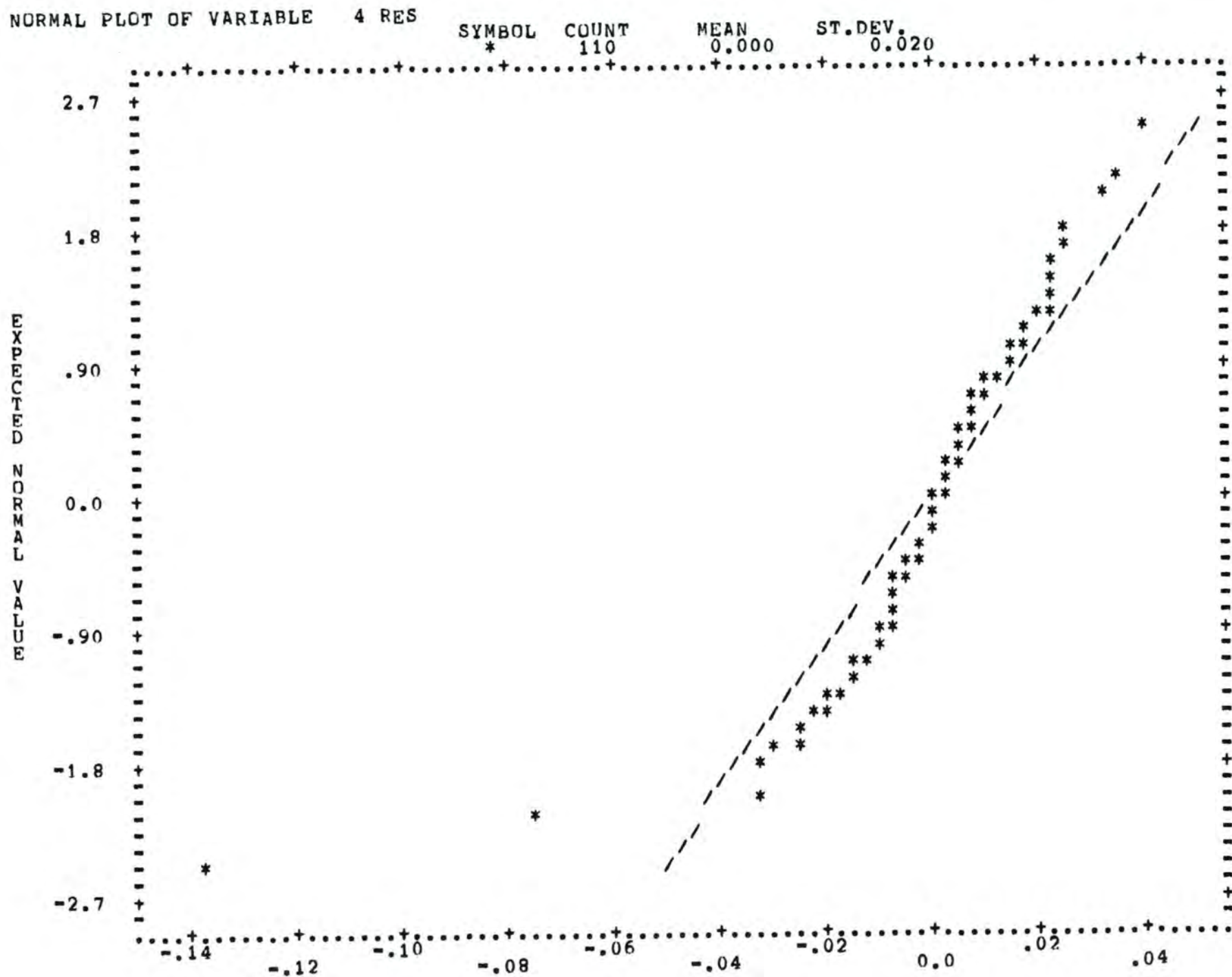


Figure 5.5 : Graphique des erreurs en fonction des valeurs transformées de la variable TAUXMEM.



Figure 5.6 : Graphique normal des erreurs pour la variable de performance TAUXMEM.



### 5.3.2 Méthodologie utilisée :

Pour chaque variable à expliquer, nous présentons le meilleur modèle linéaire normal que nous avons obtenu au moyen d'un logiciel statistique disponible sur le VAX (le "GLIM"). Pour choisir ce modèle, nous avons adopté la démarche suivante :

- \* Transformation éventuelle de la variable à expliquer s'il s'agit d'un taux :  $Y \rightarrow \ln(Y / (1 - Y))$ .

Ceci permet de travailler avec une variable non plus comprise entre 0 et 1 mais entre  $-\infty$  et  $+\infty$ .

- \* Exécution d'un modèle linéaire normal M1.

Ce modèle nécessite en entrée :

- les valeurs de la variable de performance éventuellement transformée
- les valeurs des variables explicatives susceptibles d'influencer la variable de performance étudiée.

Il fournit notamment comme résultat :

- la valeur de la somme des carrés des erreurs : SCE1.
- le degré de liberté :  $n - \text{rang } A1$ .

- \* Exécution du modèle le plus général M.

Il s'agit du modèle de classification simple présenté à la page 146.

Il fournit notamment comme résultat :

- la valeur de la somme des carrés des erreurs : SCE.
- le degré de liberté :  $n - \text{rang } A$  (rang A est le nombre de cellules).

- \* Test de M1 par rapport à M.

Ce test permet de savoir si l'observation est défavorable au modèle M1 par rapport au modèle le plus général M qui est supposé vrai à priori.

L'hypothèse M1 sera rejetée par rapport à M, au niveau  $\alpha$  (erreur de première espèce), si l'observation est dans la région critique déterminée par la valeur de  $r(w)$  et du  $Q_F(\text{rang } A - \text{rang } A1, n - \text{rang } A) (1 - \alpha)$ .

L'hypothèse M1 peut également être rejetée au moyen de la probabilité de dépassement (voir page 147).

\* Estimation simultanée des paramètres.

Si le modèle linéaire normal  $M_1$  est rejeté, il faut dans ce cas essayer d'autres modèles :

- en ajoutant éventuellement de nouvelles variables
- en effectuant une transformation sur les variables explicatives
- en transformant la variable à expliquer
- ou en utilisant un modèle supposant d'autres hypothèses de départ (par exemple, autre distribution que la normale).

Par contre, si le modèle  $M_1$  est accepté, on estime simultanément les variables explicatives afin de savoir si elles sont significatives ou non. Lorsqu'on remarque que certaines d'entre elles ne le sont pas, on teste un autre modèle linéaire normal identique à celui utilisé la première fois mais en ne tenant plus compte de ces variables. Ce modèle sera celui que nous présenterons dans la suite du développement.



### 5.3.3 Résultats de modélisation :

#### \* Taux d'utilisation des processeurs de lignes : TAUXPL.

Vu le nombre important de variables explicatives, il nous a semblé nécessaire d'effectuer une analyse des corrélations afin de faciliter l'interprétation des résultats ultérieurs. Ceci nous a permis de supprimer deux variables qui étaient fortement corrélées avec d'autres. Il s'agit du nombre de portefeuille par processeur de lignes (P) et du nombre de ligne (G). Cette élimination ne fausse pas les résultats statistiques car ce sont deux types d'utilisateur employant très rarement les services offerts par le réseau. De plus, nous avons remarqué que durant toute la période de mesure, ils envoyaient très peu de segments.

Remarquons que pour cette variable à expliquer, nous avons été contraints de diviser les données en quatre catégories (types d'application front-ends, agences se trouvant sur un noeud avec front-end, agences se trouvant sur un noeud sans front-end et bureautique) car aucun modèle n'était accepté lorsqu'on les laissait en un seul groupe. En effet, certaines variables explicatives influençaient les résultats relatifs à un type d'application mais étaient totalement inappropriées à un autre type.

#### \* \* TYPE D'APPLICATION FRONT-ENDS \* \*

##### a) Démarche suivie :

##### - Transformation de la variable à expliquer.

TAUXPL ---->  $\ln (\text{TAUXPL} / (1 - \text{TAUXPL}))$ .

##### - Sélection des variables explicatives.

Nous avons choisi de travailler avec les variables suivantes :

- Nombre d'agences reliées au front-end (AF)
- Nombre de bureautiques se trouvant sur le processeur étudié et utilisant les services du réseau durant la période de la mesure (B)
- Nombre d'agences sur le même processeur que le front-end dont elles dépendent (A)
- Nombre d'agences se trouvant sur le processeur étudié et ne dépendant pas du front-end situé sur celui-ci (AMP).

##### - Exécution du modèle linéaire normal M1.

Il s'agit du modèle utilisant les variables explicatives retenues au point précédent (AF, B, A et AMP).

##### - Exécution du modèle le plus général M.

- Acceptation du modèle M1 par rapport au modèle général M.

$$\text{En effet, il fournissait un } r(w) = \frac{((7.647 - 6.970) / 5)}{(6.970 / 38)}$$

c'est-à-dire 0.7381 inférieur au  $Q_F(5, 38)$  (0.95) (égal à plus ou moins 2.466). La probabilité de dépassement confirme également cette acceptation.

- Estimation simultanée des paramètres.

Cette estimation révélait que les quatre variables explicatives utilisées dans le modèle étaient toutes significatives et intervenaient donc dans le modèle retenu.

#### b) Interprétation des résultats :

L'exploitation statistique des résultats nous permet de dire que durant la nuit, le pourcentage d'utilisation des processeurs de lignes est influencé essentiellement par quatre variables.

Il s'agit du :

- **Nombre d'agences reliées au front-end situé sur le processeur étudié (AF).**

Plus ce nombre est élevé, plus le front-end devra envoyer de données de mises à jour à ses agences respectives. Ce qui entraînera une très grande utilisation du processeur de lignes sur lequel se situe le front-end.

- **Nombre de bureautiques situées sur le processeur étudié (B).**

Cette variable peut influencer considérablement la charge d'un processeur durant la période réservée uniquement au type d'application front-end. En effet, lors de la détermination graphique des périodes de charge, nous avons remarqué que la bureautique et les front-ends envoyaient simultanément des segments en fin de soirée.

- **Nombre d'agences sur le même processeur que le front-end dont elles dépendent (A).**

Plus ce nombre est élevé, plus il y aura de trafic à l'intérieur du processeur. En effet, celui-ci devra non seulement gérer l'envoi des données de mises à jour du front-end vers toutes ses agences, mais également la réception des informations des agences (du type A) qui lui sont reliées et qui se trouvent sur le même processeur.



- Nombre d'agences se trouvant sur le processeur étudié et ne dépendant pas du front-end situé sur celui-ci (AMP).

Pour ce type d'agence, le processeur devra uniquement gérer la réception des données de mises à jour envoyées par leur front-end respectif.

## \* \* TYPE D'APPLICATION AGENCES \* \*

### \* \* \* Agences se trouvant sur un noeud avec front-end \* \* \*

#### a) Démarche suivie :

- - - - -

- Transformation de la variable à expliquer.

TAUXPL ---->  $\ln (\text{TAUXPL} / (1 - \text{TAUXPL}))$ .

- Sélection des variables explicatives.

Nous avons choisi de travailler avec les variables suivantes :

- Nombre maximum de circuits virtuels ouverts permis à un instant donné (BU)
- Nombre d'agences reliées au front-end (AF)
- Nombre d'agences sur le même processeur que le front-end dont elles dépendent (A)
- Nombre d'agences se trouvant sur le processeur étudié et ne dépendant pas du front-end situé sur celui-ci (AMP).

- Exécution du modèle linéaire normal M1.

Il s'agit du modèle utilisant les variables explicatives retenues au point précédent (BU, AF, A et AMP).

- Exécution du modèle le plus général M.

- Acceptation du modèle M1 par rapport au modèle général M.

En effet, il fournissait un  $r(w) = \frac{((2.509 - 2.354) / 1)}{(2.354 / 30)}$

c'est-à-dire 1.9753 inférieur au  $Q_F(1, 30)$  (0.95) (égal à plus ou moins 4.17). La probabilité de dépassement confirme également cette acceptation.

- Estimation simultanée des paramètres.

Cette estimation révélait que les quatre variables explicatives utilisées dans le modèle étaient toutes significatives et intervenaient donc dans le modèle retenu.



b) **Interprétation des résultats :**  
 - - - - -

L'exploitation statistique des résultats nous permet de dire que durant la journée, le pourcentage d'utilisation des processeurs de lignes pour les agences se trouvant sur un noeud avec front-end est influencé essentiellement par quatre variables.

Il s'agit du :

- **Nombre maximum de circuits virtuels ouverts permis à un instant donné (BU).**

L'interprétation de cette variable nous semble difficile à priori; il nous paraît nécessaire d'effectuer une analyse complémentaire afin de mesurer l'impact réel de cette variable sur l'utilisation d'un processeur de lignes. Nous pensons toutefois qu'elle joue un rôle non négligeable car elle dépend du nombre et du type des utilisateurs reliés à un processeur de lignes déterminé et qu'elle est donc susceptible d'influencer la charge de celui-ci.

- **Nombre d'agences reliées au front-end situé sur le processeur étudié (AF).**

Plus ce nombre est élevé, plus le front-end recevra de données en provenance des agences de ce type. Ce qui entraînera une très grande utilisation du processeur de lignes sur lequel se situe le front-end.

- **Nombre d'agences sur le même processeur que le front-end dont elles dépendent (A).**

Plus ce nombre est élevé, plus il y aura de trafic à l'intérieur du processeur. Celui-ci devra gérer en même temps l'envoi des informations encodées par les employés des agences de ce type et la réception des données par le front-end.

- **Nombre d'agences se trouvant sur le processeur étudié et ne dépendant pas du front-end situé sur celui-ci (AMP).**

Pour ce type d'agence, le processeur devra uniquement gérer l'envoi des informations encodées par les employés de ces agences vers leur front-end respectif.

\* \* \* Agences se trouvant sur un noeud sans front-end \* \* \*

a) Démarche suivie :

- - - - -

- Transformation de la variable à expliquer.

TAUXPL ---->  $\ln (TAUXPL / (1 - TAUXPL))$ .

- Sélection des variables explicatives.

Nous avons choisi de travailler avec les variables suivantes :

- Nombre maximum de circuits virtuels ouverts permis à un instant donné (BU)
- Nombre d'agences sur le processeur étudié (AL)
- Configuration du processeur. Nous avons uniquement retenu les types d'utilisateurs autres que les agences, susceptibles d'établir des circuits virtuels durant la journée.  
Il s'agit du :

- Nombre de bureaux se trouvant sur le processeur étudié et utilisant les services du réseau durant la période de la mesure (B)
- Nombre de "CCNC" (C)
- Nombre de front-ends de test (T)
- Nombre de "RJ2TEST" (J)
- Nombre de "CALLO1" (I)
- Nombre d'"HQR80" (HR)
- Nombre d'"HQC81" (HC)
- Nombre de salles de cours (S)
- Nombre d'"INTERCER" (IN).

- Exécution du modèle linéaire normal M1.

Il s'agit du modèle utilisant les variables explicatives retenues au point précédent (BU, AL, B, C, T, J, I, HR, HC, S et IN).

- Exécution du modèle le plus général M.

- Acceptation du modèle M1 par rapport au modèle général M.

En effet, il fournissait un  $r(w) = \frac{((5.612 - 4.351) / 12)}{(4.351 / 72)}$

c'est-à-dire 1.73 inférieur au  $Q_F(12, 72)$  (0.95) (égal à plus ou moins 1.902). La probabilité de dépassement est très petite, ce qui devrait motiver le recours à une analyse plus fine.



- Estimation simultanée des paramètres.

Cette estimation révélait que les onze variables explicatives utilisées dans le modèle étaient toutes significatives et intervenaient donc dans le modèle retenu.

En outre, nous avons essayé de regrouper certaines variables explicatives, mais le modèle n'était plus accepté, ce qui signifie que toutes les variables retenues ont leur importance.

b) Interprétation des résultats :

L'exploitation statistique des résultats nous permet de dire que durant la journée, le pourcentage d'utilisation des processeurs de lignes pour les agences se trouvant sur un noeud sans front-end est influencé essentiellement par onze variables.

Il s'agit du :

- **Nombre maximum de circuits virtuels ouverts permis à un instant donné (BU).**

Voir remarque page 157.

- **Nombre d'agences reliées au processeur étudié (AL).**

Plus ce nombre est élevé, plus le processeur devra gérer l'envoi d'informations vers les différents front-ends dont dépendent les agences de ce type. Tous les segments envoyés transiteront donc à travers le réseau car aucun noeud étudié dans ce type d'application ne comporte de front-end.

- **Configuration du processeur (B, C, T, J, I, HR, HC, S et IN).**

Durant la journée, l'utilisation d'un processeur de lignes dépend fortement de sa configuration, c'est-à-dire du nombre et du type des utilisateurs susceptibles de l'utiliser.

\* \* TYPE D'APPLICATION BUREAUTIQUE \* \*

a) Démarche suivie :

- Transformation de la variable à expliquer.

TAUXPL ---->  $\ln (TAUXPL / (1 - TAUXPL))$ .



- Sélection des variables explicatives.

Nous avons choisi de travailler avec les variables suivantes :

- Nombre maximum de circuits virtuels ouverts permis à un instant donné (BU)
- Nombre de "call01" sur le processeur étudié (I)
- Nombre d'"intercer" sur le processeur étudié (IN).

Ces deux derniers types d'utilisateur sont en fin de soirée, les principaux consommateurs du réseau.

- Exécution du modèle linéaire normal M1.

Il s'agit du modèle utilisant les variables explicatives retenues au point précédent (BU, I et IN).

- Exécution du modèle le plus général M.

- Acceptation du modèle M1 par rapport au modèle général M.

En effet, il fournissait un  $r(w) = \frac{((4.973 - 3.589) / 3)}{(3.589 / 21)}$

c'est-à-dire 2.6993 inférieur au  $Q_F(3, 21)$  (0.95) (égal à plus ou moins 3.07). La probabilité de dépassement confirme cette acceptation.

- Estimation simultanée des paramètres.

Cette estimation révélait que les trois variables explicatives utilisées dans le modèle étaient toutes significatives et intervenaient donc dans le modèle retenu.

b) Interprétation des résultats :

L'exploitation statistique des résultats nous permet de dire qu'en fin de soirée, le pourcentage d'utilisation des processeurs de lignes est influencé essentiellement par trois variables.

Il s'agit du :

- Nombre maximum de circuits virtuels ouverts permis à un instant donné (BU).

Voir remarque page 157.

- Nombre de "call01" sur le processeur étudié (I).

Il s'agit d'un type d'utilisateur influençant la charge du réseau de manière constante. Il est dès lors normal qu'il influence considérablement le pourcentage d'utilisation du processeur de lignes sur lequel il se trouve et principalement en fin de

soirée lorsque la charge fournie par les principaux consommateurs est relativement faible.

- **Nombre d'"intercer" sur le processeur étudié (IN).**

Ce type d'utilisateur influence le pourcentage d'utilisation du processeur de lignes sur lequel il se trouve car nous avons remarqué qu'il travaillait considérablement en fin de soirée.

\* Disponibilité de la mémoire des processeurs de lignes : TAUXMEM.

Même remarque que pour le taux d'utilisation des processeurs de lignes en ce qui concerne la corrélation de leurs variables explicatives (voir page 154).

a) **Démarche suivie :**

- - - - -

- Transformation de la variable à expliquer.

TAUXMEM ----> ln (TAUXMEM).

- Sélection des variables explicatives.

Nous avons choisi de travailler avec les variables suivantes :

- Type d'application (AP)
- Nombre maximum de circuits virtuels ouverts permis à un instant donné (BU)
- Nombre d'agences reliées au front-end (AF)
- Configuration du processeur :
  - Nombre d'agences sur le processeur étudié (AL)
  - Nombre de bureaux se trouvant sur le processeur étudié et utilisant les services du réseau durant la période de la mesure (B)
  - Nombre de front-ends sur le processeur (F)
  - Nombre de "CCNC" (C)
  - Nombre de front-ends de test (T)
  - Nombre de "RJ2TEST" (J)
  - Nombre de "CALLO1" (I)
  - Nombre d'"HQR80" (HR)
  - Nombre d'"HQC81" (HC)
  - Nombre de salles de cours (S)
  - Nombre d'"INTERCER" (IN).

- Exécution du modèle linéaire normal M1.

Il s'agit du modèle utilisant les variables explicatives retenues au point précédent :

(AP, BU, AF, AL, B, F, C, T, J, I, HR, HC, S et IN).

- Exécution du modèle le plus général M.



- Acceptation du modèle M1 par rapport au modèle général M.

((0.0455 - 0.02741) / 34)

En effet, il fournissait un  $r(w) = \frac{((0.0455 - 0.02741) / 34)}{(0.02741 / 60)}$

c'est-à-dire 1.1646 inférieur au  $Q_F(34, 60)$  (0.95) (égal à plus ou moins 1.626). La probabilité de dépassement confirme cette acceptation.

- Estimation simultanée des paramètres.

Cette estimation révélait que les quatorze variables explicatives utilisées dans le modèle étaient toutes significatives et intervenaient donc dans le modèle retenu.

b) Interprétation des résultats :

L'exploitation statistique des résultats nous permet de dire que la disponibilité de la mémoire des processeurs de lignes est influencée essentiellement par quatorze variables.

Il s'agit du :

- **Type d'application (AP).**

Le type d'application influence considérablement la disponibilité de la mémoire. En effet, lorsque les front-ends mettent à jour les bases de données de leurs agences, elles ouvrent beaucoup de circuits virtuels. L'établissement de ces derniers nécessitent donc l'utilisation d'une grande place mémoire. Il en est de même pour les deux autres types d'application, mais dans une proportion plus limitée.

- **Nombre maximum de circuits virtuels ouverts permis à un instant donné (BU).**

Même remarque que page 157, bien qu'il s'agisse ici de la mémoire des processeurs de lignes.

- **Configuration du processeur :**

(AF, AL, B, F, C, T, J, I, HR, HC, S et IN).

Chaque type d'utilisateur d'un processeur de lignes utilise une place mémoire qui lui est propre et nous avons remarqué que chacun d'eux avait une importance non négligeable; en effet, comme pour le taux d'utilisation d'un processeur de lignes, nous n'avons pas pu les regrouper.



- \* Pourcentage d'utilisation des troncs (TAUXTR) et Temps d'existence d'un appel (TEMPS).

Ces deux variables de performance n'ont pas fait l'objet d'une étude statistique approfondie car les responsables du réseau étaient plus intéressés par l'étude des autres variables à expliquer. Faute de temps, nous n'avons pas pu les aborder.

- \* Nombre de circuits virtuels ouverts : CIRCUIT.

a) Démarche suivie :

- - - - -

Pour cette variable de performance, nous avons suivi les points de la démarche présentée ci-dessous :

- Exécution du modèle linéaire normal M1.

Il s'agit du modèle reprenant toutes les variables explicatives retenues dans le chapitre deux (AP, BU, AL, F et B).

- Exécution du modèle le plus général M.

- Acceptation du modèle M1 par rapport au modèle général M.

En effet, il fournissait un  $r(w) = \frac{((496,8 - 281) / (83 - 45))}{(281 / 45)}$

c'est-à-dire 0.9094 inférieur au  $Q_F(38,45)$  (0.95) (égal à plus ou moins 1.6755). La probabilité de dépassement confirme cette acceptation.

- Estimation simultanée des paramètres.

Cette estimation révélait que les variables explicatives (BU, F, B) n'étaient pas significatives.

- Essai d'un autre modèle linéaire normal M2.

Nous avons essayé un modèle reprenant uniquement les variables explicatives significatives retenues au point précédent. Il s'agit du type d'application (AP) et du nombre d'agences sur le processeur de lignes étudié (AL).

- Acceptation du modèle réduit M2 par rapport au modèle général M.

En effet, il fournissait un  $r(w) = \frac{((523,4 - 281) / (86 - 45))}{(281 / 45)}$

c'est-à-dire 0.9467 inférieur au  $Q_F(41,45)$  (0.95) (égal à plus ou moins 1.6624). La probabilité de dépassement confirme cette acceptation.

b) **Interprétation des résultats :**

- - - - -

L'exploitation statistique des résultats nous permet d'admettre que le nombre de circuits virtuels ouverts par processeur de lignes à un instant donné est influencé essentiellement par deux variables.

Il s'agit du :

- **Type d'application (AP).**

En début de soirée, lorsque l'application principale est la bureautique, le nombre de circuits virtuels ouverts est relativement faible. Ceci se comprend aisément par le fait que très peu de bureautiques sur un même processeur parlent en même temps.

Durant la nuit, ce nombre devient considérable. En effet, les front-ends ouvrent pendant cette période un nombre élevé de circuits virtuels afin de mettre à jour les bases de données de leurs agences respectives.

Durant la journée, ce nombre diminue relativement. C'est la période durant laquelle les agences envoient vers le front-end dont elles dépendent, les informations encodées par les employés. Il est à noter que toutes les agences n'effectuent pas ce genre d'opération en même temps. Si tel était le cas, la variable étudiée devrait prendre de plus grandes valeurs.

- **Nombre d'agences reliées au processeur de lignes étudié (AL).**

Le nombre de circuits virtuels ouverts pour un processeur de lignes déterminé dépend directement du nombre d'agences qui lui sont reliées.



\* Pourcentage de segments locaux à un noeud : TAUXSEGLOC.

Pour cette variable à expliquer, nous avons été contraints de diviser les données en trois catégories (types d'application front-ends, agences et bureautique) car aucun modèle n'était accepté lorsqu'on les laissait en un seul groupe. En effet, certaines variables explicatives influençaient les résultats relatifs à un type d'application mais étaient totalement inappropriées à un autre type.

\* \* TYPE D'APPLICATION FRONT-ENDS \* \*

a) Démarche suivie :

- - - - -

- Transformation de la variable à expliquer.

TAUXSEGLOC ---->  $\ln (\text{TAUXSEGLOC} / (1 - \text{TAUXSEGLOC}))$ .

- Sélection des variables explicatives.

Nous avons choisi de travailler avec les variables suivantes :

- Nombre d'agences se trouvant sur le même noeud que le front-end dont elles dépendent (AFN)
- Nombre d'agences reliées à un autre noeud que celui étudié et dépendant du front-end situé sur le noeud analysé (AFPN)
- Interaction entre la variable AFN et la variable AFPN (AGLF).

- Exécution du modèle linéaire normal M1.

Il s'agit du modèle reprenant les variables explicatives retenues au point précédent (AFN, AFPN et AGLF).

- Exécution du modèle le plus général M.

- Acceptation du modèle M1 par rapport au modèle général M.

En effet, il fournissait un  $r(w) = \frac{((0.3096 - 0.2796) / 2)}{(0.2796 / 12)}$

c'est-à-dire 0.6437 inférieur au  $Q_F(2, 12)$  (0.95) (égal à plus ou moins 3.89). La probabilité de dépassement confirme cette acceptation.

- Estimation simultanée des paramètres.

Cette estimation révélait que les trois variables explicatives utilisées dans le modèle étaient toutes significatives et intervenaient donc dans le modèle retenu.



b) **Interprétation des résultats :**

L'exploitation statistique des résultats nous permet de dire que durant la nuit, le pourcentage de segments locaux à un noeud est influencé essentiellement par trois variables.

Il s'agit du :

- **Nombre d'agences reliées au noeud étudié et dépendant du front-end situé sur le même noeud (AFN).**

Plus ce nombre est élevé par rapport à la valeur de AFN, plus le pourcentage de segments envoyés durant la nuit par le front-end vers les agences de ce type sera considérable.

- **Nombre d'agences reliées à un autre noeud que celui étudié et dépendant du front-end situé sur le noeud analysé (AFPN).**

Plus ce nombre est élevé par rapport à la valeur de AFN sur un noeud déterminé, plus le pourcentage de segments locaux envoyés durant la nuit sera faible, puisque la majeure partie des données de mise à jour transiteront à travers le réseau afin d'arriver dans des agences d'un autre noeud.

- **Interaction entre la variable AFN et la variable AFPN (AGL).**

L'interprétation de cette variable est assez difficile; elle nous semble nécessaire pour faire intervenir un certain pourcentage entre le nombre d'agences du type AFN et le nombre d'agences du type AFPN.

\* \* TYPE D'APPLICATION AGENCES \* \*a) **Démarche suivie :**

- Transformation de la variable à expliquer.

TAUXSEGLOC ---->  $\ln (\text{TAUXSEGLOC} / (1 - \text{TAUXSEGLOC}))$ .

- Sélection des variables explicatives.

Nous avons choisi de travailler avec les variables suivantes :

- Nombre de front-ends sur le noeud (**FEN**)
- Existence d'agences "mère" et d'agences "filles" sur le noeud (**AGENMF**)
- Pourcentage d'agences se trouvant sur le noeud étudié et dépendant du front-end sur le même noeud par rapport au nombre total d'agences situées sur le noeud (**AGMF**)

- Exécution du modèle linéaire normal M1.

Il s'agit du modèle reprenant les variables explicatives retenues au point précédent (FEN, AGENMF et AGMF).

- Exécution du modèle le plus général M.

- Acceptation du modèle M1 par rapport au modèle général M.

En effet, il fournissait un  $r(w) = \frac{((8.707 - 7.192) / 12)}{(7.192 / 16)}$

c'est-à-dire 0.28 inférieur au  $Q_F(12, 16)$  (0.95) (égal à plus ou moins 2.42). La probabilité de dépassement confirme cette acceptation.

- Estimation simultanée des paramètres.

Cette estimation révélait que les trois variables explicatives utilisées dans le modèle étaient toutes significatives et intervenaient donc dans le modèle retenu.

b) Interprétation des résultats :

L'exploitation statistique des résultats nous permet de dire que durant la journée, le pourcentage de segments locaux à un noeud est influencé essentiellement par trois variables.

Il s'agit du :

- **Nombre de front-ends situés sur le noeud (FEN).**

Lorsqu'un noeud ne possède pas de front-end, le pourcentage de segments locaux durant la journée est quasi nul car tous les segments doivent transiter à travers le réseau pour aboutir au front-end approprié. Dans le cas contraire, tous les segments en provenance des agences reliées au même noeud que le front-end dont elles dépendent seront internes au noeud.

- **Présence d'agences "mère" et d'agences "filles" sur le noeud (AGENMF)**

L'existence d'agences de ce type peut influencer le pourcentage de segments locaux du noeud sur lequel elles se trouvent, puisque toutes les données transmises entre elles ne transitent pas à travers le réseau mais restent internes au noeud.

- Pourcentage d'agences se trouvant sur le noeud étudié et dépendant du front-end sur le même noeud (AFN) par rapport au nombre total d'agences situées sur le noeud (AFN + AAF) = AGMF.

Plus ce rapport est grand, plus le pourcentage de segments locaux sera considérable car la totalité des segments envoyés par les agences de type (AFN) restera interne au noeud.

\* \* TYPE D'APPLICATION BUREAUTIQUE \* \*

Les données relatives à ce type d'application n'ont pas fait l'objet d'une étude statistique car elles avaient toujours une valeur constante d'un jour à l'autre. En effet, le pourcentage de segments locaux à un noeud était toujours nul sauf sur le noeud de Bruxelles (80) où il était égal à plus ou moins 95%.



\* Disponibilité du réseau : TAUXDIS.

Pour cette variable à expliquer, nous avons été contraints de diviser les données en trois catégories (types d'application front-ends, agences et bureautique) car aucun modèle n'était accepté lorsqu'on les laissait en un seul groupe. En effet, certaines variables explicatives influençaient les résultats relatifs à un type d'application mais étaient totalement inappropriées à un autre type.

\* \* TYPE D'APPLICATION FRONT-ENDS \* \*

a) Démarche suivie :

- - - - -

- Transformation de la variable à expliquer.

TAUXDIS ---->  $\ln (\text{TAUXDIS} / (1 - \text{TAUXDIS}))$ .

- Sélection des variables explicatives.

Nous avons choisi de travailler avec les variables représentant les principaux destinataires des appels en provenance des front-ends.

Il s'agit du :

- Nombre d'agences se trouvant sur le même noeud que le front-end dont elles dépendent (**AFN**)
- Nombre d'agences reliées à un autre noeud que celui étudié et dépendant du front-end situé sur le noeud analysé (**AFPN**)
- Nombre d'agences reliées au noeud étudié et dépendant d'un front-end non situé sur ce noeud (**AAF**)
- Pourcentage d'agences situées sur le noeud par rapport au nombre total d'agences sur le noeud et d'agences dépendant du front-end sur un autre noeud (**AGN**).

- Exécution du modèle linéaire normal M1.

Il s'agit du modèle reprenant les variables explicatives retenues au point précédent (AFN, AFPN, AAF et AGN).

- Exécution du modèle le plus général M.

- Acceptation du modèle M1 par rapport au modèle général M.

((35.53 - 32.59) / 1)

En effet, il fournissait un  $r(w) = \frac{\text{-----}}{(32.59 / 24)}$

c'est-à-dire 2.165 inférieur au  $Q_{F(1,24)}(0.95)$  (égal à plus ou moins 4.26). La probabilité de dépassement est faible et nécessiterait la recherche éventuelle d'autres modèles.

- Estimation simultanée des paramètres.

Cette estimation révélait que les quatre variables explicatives utilisées dans le modèle étaient toutes significatives et intervenaient donc dans le modèle retenu.

b) Interprétation des résultats :

Une simple analyse des résultats nous montre que durant la nuit, la disponibilité du réseau à chaque noeud est très grande. Elle est de l'ordre de plus ou moins 85%. On y rencontre rarement un problème. L'exploitation statistique des résultats rend responsables les différentes agences (destinataires des appels) des difficultés pouvant intervenir dans le réseau. En effet, nous avons remarqué que le premier code d'appel refusé à un noeud correspondait à un service inactif à l'extrémité de l'appel. Il est dès lors normal que les agences de différents types influencent la disponibilité du réseau. Plus le nombre d'agences intéressées par les mises à jour est grand, plus grande est la possibilité de rencontrer un problème. Une étude détaillée pourrait nous donner plus d'information à ce sujet.

\* \* TYPE D'APPLICATION AGENCES \* \*

a) Démarche suivie :

- Transformation de la variable à expliquer.

TAUXDIS ---->  $\ln (\text{TAUXDIS} / (1 - \text{TAUXDIS}))$ .

- Sélection des variables explicatives.

Nous avons choisi de travailler avec les variables suivantes :

- Nombre de front-ends sur le noeud (FEN)
- Nombre d'agences reliées à un autre noeud que celui étudié et dépendant du front-end situé sur le noeud analysé (AFPN)
- Nombre d'agences reliées au noeud étudié et dépendant d'un front-end non situé sur ce noeud (AAF)
- Nombre d'agences reliées au noeud étudié (AGEN = AFN + AAF).

- Exécution du modèle linéaire normal M1.

Il s'agit du modèle reprenant les variables explicatives retenues au point précédent (FEN, AFPN, AAF et AGEN).

- Exécution du modèle le plus général M.



- Acceptation du modèle M1 par rapport au modèle général M.

En effet, il fournissait un  $r(w) = \frac{((257.5 - 179.9) / (11))}{(179.9 / 48)}$

c'est-à-dire 1.8822 inférieur au  $Q_F(11, 48)$  (0.95) (égal à plus ou moins 2.006). La probabilité de dépassement est faible et nécessiterait la recherche éventuelle d'autres modèles.

- Estimation simultanée des paramètres.

Cette estimation révélait que certaines variables explicatives (AFPN, AAF et AGEN) n'étaient pas significatives.

- Essai d'un autre modèle linéaire réduit M2.

Nous avons essayé un autre modèle linéaire avec comme unique variable explicative le nombre de front-ends sur le noeud (FEN).

- Acceptation du modèle réduit M2 par rapport au modèle général M.

En effet, il fournissait un  $r(w) = \frac{((262.1 - 179.9) / (14))}{(179.9 / 48)}$

c'est-à-dire 1.5665 inférieur au  $Q_F(14, 48)$  (0.95) (égal à plus ou moins 1.9146). La probabilité de dépassement est faible et nécessiterait la recherche éventuelle d'autres modèles.

b) Interprétation des résultats :

Une simple analyse des résultats nous montre que durant la journée, la disponibilité du réseau à chaque noeud est très grande. Elle est de l'ordre de plus ou moins 99%. On y rencontre rarement un problème. L'exploitation statistique des résultats rend responsables les différents front-ends (destinataires des appels) des difficultés pouvant intervenir dans le réseau. En effet, nous avons remarqué que le premier code d'appel refusé à un noeud correspondait à un service inactif à l'extrémité de l'appel. Il est dès lors normal que les front-ends influencent la disponibilité du réseau. Lorsqu'un noeud possède un front-end, il est probable que celui-ci ait un problème. Une étude détaillée pourrait nous donner plus d'information à ce sujet.

\* \* TYPE D'APPLICATION BUREAUTIQUE \* \*.

Les données relatives à ce type d'application n'ont pas fait l'objet d'une étude statistique approfondie. Néanmoins, nous n'avons jamais observé une disponibilité du réseau inférieure à 75 %.



### CONCLUSIONS.

Nous voilà arrivés au terme de notre étude et il convient maintenant d'essayer d'en tirer quelques conclusions.

La manière dont nous avons conduit notre travail nous amène à plusieurs réflexions :

- tout d'abord, nous avons conscience qu'une étude statistique telle que celle que nous avons présentée dans ce travail, devrait aller de pair avec une autre étude éventuellement statistique, de manière à soit confirmer nos propres aboutissements, soit les remettre en question, auquel cas nous serions contraints d'en rechercher les causes.

Dans le cadre de ce mémoire, nous avons dans certains cas effectué en parallèle une analyse en composantes principales, mais les résultats que nous avons obtenus engendraient une interprétation difficile des axes principaux (retenus selon le pourcentage de variance que l'on désirait expliquer), de même que des points projetés sur ceux-ci. Malheureusement, nous n'avons pas pu creuser davantage le sujet, faute de temps.

D'autre part, nous sommes également conscients qu'il faut que les mesures effectuées dans le but d'une interprétation statistique des résultats obtenus soient en nombre suffisant pour pouvoir réaliser une inférence correcte, sans toutefois que leur volume ne soit trop grand de manière à ne pas gêner leur interprétation. Dans certains cas, nous nous sommes trouvés confrontés à ce problème; nous pensons tout spécialement aux mesures concernant les

front-ends qui ne sont qu'au nombre de six sur l'ensemble du réseau.

Nous proposons au point suivant un moyen de répondre à ce problème.

- ensuite, nous sommes convaincus que le travail réalisé ici ne représente qu'une première étape dans l'étude du réseau de la Générale de Banque. Nous suggérons qu'une étude plus fine soit réalisée séparément pour chaque variable à expliquer, mais ceci nécessiterait un accroissement des moyens qui étaient mis à notre disposition. L'avantage d'une telle démarche est qu'elle supprimerait l'hypothèse restrictive que nous avons adoptée au chapitre trois (Il s'agissait d'imputer toute la charge d'un processeur à un seul type d'application à l'intérieur d'une plage horaire). Par exemple, il serait tout à fait possible d'étudier l'impact réel d'un front-end sur le pourcentage d'utilisation d'un processeur de lignes en reconfigurant de plusieurs manières ce processeur et en observant le comportement du réseau (ou du processeur) lors de ces différents changements.

Pour les **agences**, nous proposons notamment :

- d'effectuer des mesures sur un processeur supportant uniquement de telles lignes (éventuellement une seule)
- de descendre jusqu'au niveau de l'agence elle-même afin de pouvoir tenir compte de leur grandeur respective.

Le problème de la charge fournie au réseau par la **bureautique** doit déjà avoir été traité par les responsables de la Générale de Banque puisqu'à la suite des résultats



obtenus au chapitre trois, ils ont remarqué que ce type d'application fournissait au réseau une charge qui ne correspondait pas à leur attente.

Nous pensons également qu'un type d'application tel que les 'Call01' devrait aussi faire l'objet d'une étude approfondie, parce que générateurs d'une charge non négligeable à certains moments.

Nous considérons donc que notre travail ne représente pas une fin en soi, mais bien un point de départ pour une étude en profondeur d'un problème réellement complexe.

- enfin plusieurs constatations doivent être faites quant au réseau lui-même.

Dans sa configuration actuelle, il nous a paru surdimensionné par rapport à son utilisation réelle. En effet, les mesures ont été effectuées aux heures les plus chargées, alors qu'en utilisation normale, le réseau supporte une charge réellement inférieure.

La plupart des observations effectuées sur les variables décrites au chapitre deux nous confortent dans cette idée; c'est ainsi qu'en dehors des heures de charge mentionnées dans le chapitre trois pour chaque type d'application retenu, nous observons rarement un pourcentage d'utilisation des troncs et des processeurs de lignes supérieur à 15 %; dans le même ordre d'idée, le pourcentage de disponibilité de la mémoire des processeurs de lignes est rarement inférieur à 85 - 90 %. Quant au nombre de circuits virtuels ouverts à un moment donné, il est largement inférieur à la capacité offerte sur chaque



processeur et le pourcentage de disponibilité du réseau est également très élevé.

Les problèmes de performance dont nous avons eu connaissance lors de notre stage se sont présentés lors de transferts de fichiers au départ de l'HQR80; ceci est essentiellement dû à la technique de transmission utilisée qui consiste à atteindre chaque agence au départ du noeud 80, alors qu'une technique de transmission répartie serait plus appropriée.

Nous avons également remarqué que la manière selon laquelle le réseau avait été dessiné n'était pas optimale, puisque nous avons dénombré huit troncs (sur un total de trente) qui n'étaient pas utilisés par l'algorithme de routage actuel. Il s'agit uniquement de troncs à 9600 bps. Nous devons cependant préciser que certains troncs parmi ceux-là sont toutefois susceptibles d'être utilisés en cas de problème sur un autre tronc : c'est notamment le cas de celui qui relie les noeuds d'Hasselt et de Liège et qui serait utilisé en cas de problème quelconque sur le tronc reliant les noeuds d'Hasselt et de Louvain. Ceci provient du fait qu'en supprimant ledit tronc, il devient impossible d'atteindre le noeud d'Hasselt par au moins deux chemins différents au départ d'un noeud quelconque du réseau.

Par contre, même en supprimant le tronc reliant les noeuds de Gand et de Mons, on aurait toujours la possibilité d'atteindre ces deux noeuds par deux chemins différents au départ d'un noeud quelconque du réseau. Les troncs ayant cette caractéristique sont au nombre de six.

L'ensemble de ces considérations devra bien sûr être revu puisqu'un nouvel algorithme de routage doit bientôt être

utilisé, algorithme qui tiendra compte de la charge au moment du choix du chemin utilisé. Rappelons qu'actuellement l'algorithme de routage ne tient compte que des poids accordés à chaque tronc en fonction de leur vitesse de transmission.

Il nous semble nécessaire d'attirer l'attention du lecteur sur le fait que, dans l'avenir, toutes les remarques effectuées dans cette conclusion devront être corrigées et que les divers résultats que nous avons mentionnés dans ce mémoire ne seront plus d'application.

En effet, l'utilisation réelle du réseau sera fondamentalement modifiée dans le sens où la Générale de Banque projette de ramener le nombre de front-ends batch de six à trois et qu'elle utilise depuis juillet 1986 des front-ends interactifs. De plus, les liaisons actuelles entre front-ends se réalisent indépendamment du réseau (à part une) et seront à l'avenir supportées par celui-ci. Enfin, les agences qui, jusqu'à présent, ne pouvaient pas communiquer entre elles, devraient être en mesure de le faire dans le courant de l'année 1987. Cette décision est motivée par exemple par le fait qu'actuellement, un chèque dont le compte est tenu à Arlon et qui est présenté à l'encaissement à Bruges, ne peut être directement positionné à l'agence de Bruges et nécessite dès lors l'établissement d'une communication téléphonique parallèle entre les deux agences. Selon une première estimation, le nombre moyen d'appels supplémentaires engendrés par une telle modification s'élèverait à environ sept mille par jour et représenterait une augmentation de charge sur des lignes non habituellement utilisées.

Nous souhaitons que les résultats de nos recherches soient de nature à favoriser le passage d'une utilisation à une autre et nous ne voulons pas terminer cette étude sans insister sur le fait que nous mesurons réellement la chance qui nous a été offerte d'approcher de très près l'exploitation concrète d'un réseau complexe tel que celui d'une des plus grandes sociétés de notre pays et d'avoir pu ainsi observer en pratique et en grandeur nature le comportement réel d'un réseau de transmission de données.



## TABLE DES MATIERES.

INTRODUCTION.....	Page	1
CHAPITRE 1 : DESCRIPTION DU RESEAU DE LA GENERALE DE BANQUE.....	Page	5
Résumé.....	Page	5
1.1 RESEAU A COMMUTATION PAR PAQUETS.....	Page	6
- 1.1.1 Principes.....	Page	6
1.1.2 Accès au réseau à commutation par paquets.....	Page	7
1.1.3 Avantages des réseaux à commutation par paquets.....	Page	10
1.2 RESEAU SL-10 INSTALLE A LA GENERALE DE BANQUE.....	Page	10
1.3 DESCRIPTION D'UN NOEUD SL-10.....	Page	17
1.3.1 Hardware d'un noeud SL-10.....	Page	17
1.3.2 Software d'un noeud SL-10.....	Page	25
1.4 CONTROLE DU RESEAU.....	Page	28
1.4.1 Noeud NCC (Network Control Center)....	Page	29
1.4.2 DCC (Data Control Center).....	Page	31
1.5 DESCRIPTION DES FICHIERS FOURNIS PAR LA PROCEDURE 'DAILY'.....	Page	36
1.5.1 Fichier de comptabilité des appels : 'ACNT.DAT'.....	Page	36
1.5.2 Fichier des statistiques des proces- seurs de lignes : 'LINEPROC.SRT'.....	Page	40
1.5.3 Fichier des statistiques de troncs : 'TRUNK.SRT'.....	Page	42

CHAPITRE 2 : PRESENTATION DES VARIABLES A ETUDIER.....	Page 44
Résumé.....	Page 44
2.1 VARIABLES A EXPLIQUER.....	Page 45
2.1.1 Pourcentage d'utilisation des processeurs de lignes : <b>TAUXPL</b> .....	Page 45
2.1.2 Pourcentage de disponibilité de la mémoire des processeurs de lignes : <b>TAUXMEM</b> .....	Page 45
2.1.3 Pourcentage d'utilisation des troncs : <b>TAUXTR</b> .....	Page 45
2.1.4 Temps d'existence d'un appel : <b>TEMPS</b> ..	Page 45
2.1.5 Nombre de circuits virtuels ouverts par processeur de lignes à un instant donné : <b>CIRCUIT</b> .....	Page 46
2.1.6 Pourcentage de segments locaux à un noeud : <b>TAUXSEGLOC</b> .....	Page 46
2.1.7 Disponibilité du réseau : <b>TAUXDIS</b> .....	Page 46
2.1.8 Pourcentage d'utilisation des processeurs de contrôle : <b>TAUXPC</b> .....	Page 47
2.1.9 Temps moyen entre une demande d'appel et sa confirmation : <b>TEMPSAP</b> .....	Page 47
2.2 VARIABLES EXPLICATIVES DES MESURES EFFECTUEES SUR LES NOEUDS.....	Page 47
2.2.1 Type d'application : <b>AP</b> .....	Page 47
2.2.2 Vitesse des troncs : <b>VIT</b> .....	Page 47
2.2.3 Algorithme de routage : <b>ROUT</b> .....	Page 49
2.2.4 Type du noeud origine du tronc : <b>NOEUDORI</b> .....	Page 49
2.2.5 Type du noeud destination du tronc : <b>NOEUDDES</b> .....	Page 49
2.2.6 Nombre de segments : <b>SEG</b> .....	Page 49
2.2.7 Configuration des noeuds : <b>FEN</b> et <b>AGEN</b> .....	Page 51
2.2.8 Nombre d'agences dépendant du front-end situé sur le noeud étudié : <b>AF</b> .....	Page 51
2.2.9 Nombre d'agences reliées au noeud étudié et dépendant du front-end situé sur ce noeud : <b>AFN</b> .....	Page 54
2.2.10 Nombre d'agences reliées au noeud étudié et dépendant du front-end non situé sur ce noeud : <b>AAF</b> .....	Page 54
2.2.11 Nombre d'agences reliées à un autre noeud et dépendant du front-end situé sur le noeud étudié : <b>AFPN</b> .....	Page 54
2.2.12 Présence d'agences "mère" et d'agences "fille" : <b>AGENMF</b> .....	Page 55



2.3	VARIABLES EXPLICATIVES DES MESURES EFFECTUEES SUR LES PROCESSEURS.....	Page 56
2.3.1	Configuration des processeurs de lignes.....	Page 56
2.3.2	Nombre d'agences dépendant du front-end situé sur le processeur étudié et se trouvant sur celui-ci : <b>A</b> .....	Page 64
2.3.3	Nombre d'agences se trouvant sur le processeur étudié et ne dépendant pas du front-end situé sur celui-ci : <b>AMP</b> .....	Page 65
2.3.4	Buffer disponible : <b>BU</b> .....	Page 66

### CHAPITRE 3 : DETERMINATION GRAPHIQUE DES PERIODES D'OBSERVATION..... Page 70

Résumé.....	Page 70
-------------	---------

3.1	OBJECTIF DU CHAPITRE.....	Page 71
-----	---------------------------	---------

3.2	METHODOLOGIE SUIVIE.....	Page 73
-----	--------------------------	---------

3.2.1	Comportement horaire par type d'application.....	Page 73
3.2.2	Comportement horaire pour l'ensemble du réseau.....	Page 98
3.2.3	Graphiques obtenus.....	Page 102

3.3	CONCLUSIONS.....	Page 106
-----	------------------	----------

### CHAPITRE 4 : PRISE DES MESURES DES VARIABLES A EXPLIQUER..... Page 109

Résumé.....	Page 109
-------------	----------

4.1	UTILISATION DES PROCESSEURS DE LIGNES.....	Page 110
-----	--	----------

4.1.1	Mesure de cette variable.....	Page 110
4.1.2	Prise des mesures.....	Page 111
4.1.3	Résultats obtenus.....	Page 112

4.2	DISPONIBILITE DE LA MEMOIRE DES PROCESSEURS DE LIGNES.....	Page 117
-----	---	----------

4.2.1	Mesure de cette variable.....	Page 117
4.2.2	Prise des mesures.....	Page 117
4.2.3	Résultats obtenus.....	Page 119



4.3 POURCENTAGE D'UTILISATION DES TRONCS.....	Page 122
4.3.1 Mesure de cette variable.....	Page 122
4.3.2 Prise des mesures.....	Page 123
4.3.3 Résultats obtenus.....	Page 124
4.4 TEMPS D'EXISTENCE D'UN 'CALL'.....	Page 127
4.4.1 Mesure de cette variable.....	Page 127
4.4.2 Prise des mesures.....	Page 127
4.4.3 Résultats obtenus.....	Page 130
4.5 NOMBRE DE CIRCUITS VIRTUELS OUVERTS.....	Page 132
4.5.1 Mesure de cette variable.....	Page 132
4.5.2 Prise des mesures.....	Page 132
4.5.3 Résultats obtenus.....	Page 133
4.6 POURCENTAGE DE SEGMENTS LOCAUX.....	Page 135
4.6.1 Mesure de cette variable.....	Page 135
4.6.2 Prise des mesures.....	Page 135
4.6.3 Résultats obtenus.....	Page 136
4.7 DISPONIBILITE DU RESEAU.....	Page 138
4.7.1 Mesure de cette variable.....	Page 138
4.7.2 Prise des mesures.....	Page 138
4.7.3 Résultats obtenus.....	Page 139
<b>CHAPITRE 5 : METHODE D'EXPLOITATION STATISTIQUE DES</b>	
<b>RESULTATS.....</b>	<b>Page 142</b>
Résumé.....	Page 142
5.1 INTRODUCTION.....	Page 143
5.2 LES MODELES LINEAIRES NORMAUX.....	Page 144
5.3 MODELES OBTENUS POUR CHAQUE VARIABLE DE	
PERFORMANCE.....	Page 148
5.3.1 Hypothèses de départ.....	Page 148
5.3.2 Méthodologie utilisée.....	Page 152
5.3.3 Résultats de modélisation.....	Page 154
<b>CONCLUSIONS.....</b>	<b>Page 172</b>

## BIBLIOGRAPHIE

- [ 1]- "BMDP statistical software", University of California Press, Berkeley, Los Angeles, London, 1981.
- [ 2]- Cours interne à la Générale de Banque, sur la description des fichiers de comptabilité d'appels et de statistiques, fournis par des utilitaires du réseau.
- [ 3]- **ESCHENAUER E.**, "G. Network Architecture", rapport interne, Générale de Banque, mars 1983.
- [ 4]- **MACCHI C.** et **GUILBERT J. F.**, "Transport et traitement de l'information dans les réseaux et systèmes téléinformatiques", Paris, Dunod, 1980.
- [ 5]- Manuel de référence du Datatrieve VAX11/750.
- [ 6]- Manuel de référence du GLIM (outil statistique).
- [ 7]- Manuel de référence du VAX11/750.
- [ 8]- **Noirhomme M.**, "Statistique appliquée à la gestion", cours donné à l'Institut d'Informatique, FNDP, Namur.
- [ 9]- **TANENBAUM A. S.**, "Computer Networks", Prentice - Hall International, London, 1981.
- [10]- **TONASSONE R.**, **LESQUOY E.** et **MILLIER C.**, "La régression, nouveaux regards sur une ancienne méthode statistique", Masson, Paris, 1983.